Н. И. Самойленко, А. А. Кобец

ТРАНСПОРТНЫЕ СИСТЕМЫ БОЛЬШОЙ РАЗМЕРНОСТИ

Министерство образования и науки Украины

Харьковская национальная академия городского хозяйства

Н. И. Самойленко, А. А. Кобец

ТРАНСПОРТНЫЕ СИСТЕМЫ БОЛЬШОЙ РАЗМЕРНОСТИ

Под редакцией Н.И.Самойленко

Издательство «НТМТ»

Харьков – 2010

УДК 656:519.85
С17
ББК 39.18+22.18

Самойленко М. І.
Транспортні системи великої вимірності: монографія / М. І. Самойленко, А. О. Кобець, за ред. М. І. Самойленка. – Х.: НТМТ, 2010. – 212 с. (рос. мовою).

Самойленко Н.И.
Транспортные системы большой размерности: монография / Н. И. Самойленко, А. А. Кобец, под ред. Н. И. Самойленко. – Х.: НТМТ, 2010. – 212 с.

Рекомендовано Ученым советом Харьковской национальной академии городского хозяйства в качестве монографии, протокол № 11 от 30 августа 2010 г.

Рецензенты:

А. И. Колосов – заведующий кафедрой высшей математики Харьковской национальной академии городского хозяйства, доктор физико-математических наук, профессор.

В. М. Левыкин – заведующий кафедрой информационных управляющих систем Харьковского национального университета радиоэлектроники, доктор технических наук, профессор.

Излагаются математические модели и методы решения задач по организации перевозок мелкопартионных грузов. Особое внимание уделяется транспортной задаче большой размерности и её разновидностям, а также задачам разбиения транспортной сети на заданное количество районов разработки кольцевых маршрутов.

Для научных и инженерно-технических работников научно-исследовательских, проектных и производственных организаций, связанных с проектированием и организацией транспортных перевозок.

Табл.: 11. Ил.: 63. Библиограф. наименований: 33.

ISBN 978-617-578-033-6

Содержание

Предисловие

Предметом настоящей монографии являются транспортные задачи организации перевозок мелкопартионных грузов. Другими словами, транспортные задачи большой размерности.

Основное внимание в монографии уделяется математическим и цифровым моделям оптимизационных задач, которые встречаются в инженерной практике пассажирских и грузовых перевозок. Для каждого типа транспортных задач приводятся общие содержательные и математические постановки, конкретные примеры и их решения.

Авторы сознательно выбирали для большинства задач примеры с малой размерностью транспортных сетей. Цель – усилить наглядность и упростить рассмотрение вопросов их моделирования как математического, так и цифрового.

Исключениями являются задача коммивояжера и задача разбиение транспортной сети на заданное число районов. Каждой из этих задач отводится целая глава и каждая ориентирована на большую размерность сети. Здесь на первый план выходят не столько сами модели задач, сколько методы их решения.

Монография состоит из *Введения* и четырёх разделов.

Введение является результатом совместной работы авторов. Раздел 1 написан Н. И. Самойленко. Разделы 2, 3 и 4 написаны А. А. Кобец и являются результатами её диссертационных исследований

Авторы

Введение

В последние годы во многих областях народного хозяйства для рациональной организации производства и решения производственных задач широко применяется системный подход. При таком подходе организация отдельных этапов производства должна рассматриваться с точки зрения её влияния на эффективность всего производства в целом. Организация автомобильных перевозок грузов не является исключением: комплексный характер современных исследований транспорта грузов существенно отличает их от концепции исследований прошлого века.

С точки зрения логистики, транспортные перевозки грузов представляют собой сложный процесс, который включает в себя планирование, организацию и выполнение доставки продукции, подготовку партий отправок к перевозке, организацию и проведение погрузочно-разгрузочных работ, расфасовку, упаковку, складирование товара, страхование перевозок, иногда – таможенные услуги. Важными составляющими организации транспортных перевозок являются рациональный выбор транспортных средств, наиболее полное использование грузоподъемности транспортных средств посредством правильной загрузки, соблюдение технологий при ведении погрузочно-разгрузочных работ, расчет требуемых запасов товара для обеспечения бесперебойной погрузки. В случае, когда предприятие-производитель пользуется услугами перевозчика, к элементам транспортного процесса добавляются оформление необходимых перевозочных документов, заключение договора на перевозку, расчет за услуги перевозчика. Практическая реализация всех перечисленных составляющих транспортного процесса требует соответствующих денежных и временных затрат. Естественно, эффективность перевозок и качество доставки товара будут зависеть от каждой составляющей транспортного процесса [1, 2, 3]. Только оптимальное распределение материальных, трудовых и финансовых ресурсов между всеми составляющими может гаранти-

ровать наивысшую эффективность транспортного процесса в целом.

Оптимальная организация отдельных этапов не является условием оптимальности всего процесса. Однако умение определять оптимально возможные результаты на любом этапе позволяет обеспечить исследователей и производственников важным критерием для оценки влияния этого этапа на эффективность всего процесса. Поэтому проблема оптимизации каждого этапа транспортного процесса всегда будет оставаться актуальной, но не всегда – определяющей.

Выделяют следующих основных участников процесса перевозок: производителя, экспедитора, торговую фирму и перевозчика. Следует учитывать, что в процессе перевозок каждый из участников преследует свои цели, имеет собственные доходы и затраты.

Рассмотрим доходы и затраты всех участников перевозок в рамках процесса транспортирования товара, абстрагируясь от его производства и продажи.

Доходами экспедитора являются оплата фирмой-производителем доставки товара и штрафы. Затраты экспедитора – это расходы на поиск перевозчика, оплата стоимости перевозки, расходы на доставку, штрафы и т.д.

Доходами перевозчика являются оплата тарифа перевозки, компенсации за вынужденную задержку или простоя подвижного состава. Затраты состоят из скидок экспедитору, затрат на подготовку и обеспечение перевозки.

Доходами торговой фирмы являются штрафы, оговоренные в контракте.

Модель участников системы доставки может быть представлена в виде кортежа

$$SD = \langle P, EXP, PER, TF \rangle, \qquad (0.1)$$

где $P = P(x) = \{x_1, x_2, ..., x_n\}$ – множество производителей, $EXP = EXP(y) = \{y_1, y_2, ..., y_m\}$ – множество экспедиторов, $PER = PER(z) = \{z_1, z_2, ..., z_k\}$ – множество перевозчиков;

$TF = TF(t) = \{t_1, t_2, ..., t_l\}$ – множество торговых фирм. Здесь имеет место процесс доставки, в котором принимают участие n производителей, m экспедиторов, k перевозчиков и l торговых фирм.

В общем случае целевая функция системы доставки, определяющая денежные и временные затраты на перевозку и качество доставляемого товара, имеет вид

$$E(SD) \to \min . \qquad (0.2)$$

Качество системы определяется надежностью доставки, своевременностью и продолжительностью транспортирования грузов, транспортными расходами, гибкостью перенастройки схемы транспортирования в соответствии с потребностями потребителей, сохранностью товаров, быстротой подготовки документов, затратами на крепление, загрузку и складирование грузов, расходами на страхование.

Приведенная обобщенная модель может быть использована для исследования качества системы доставки при условии существования более точных математических моделей, описывающих прибыли и затраты участников транспортного процесса. При организации процесса транспортных перевозок необходимы более конкретные модели, учитывающие все условия процесса, все требования и заинтересованность со стороны всех участников перевозок: и производителей, и экспедиторов, и перевозчиков, и получателей. В работе [4] приведен алгоритм планирования грузовых автомобильных перевозок, включающий такие модели. В общем случае алгоритм планирования предполагает выполнение следующих этапов:

1. *Формирование базы данных*. В базу данных входят сведения:

– о количестве грузоотправителей и грузополучателей;

– об объёмах груза на складах грузоотправителей и его основных характеристиках;

– о потребностях каждого грузополучателя в объёмах груза;

– о количестве транспортных средств, их типе и грузоподъемности;

– об ограничениях, накладываемых грузоотправителем и грузополучателем на партию груза, которая может быть отправлена и получена соответствующим субъектом;

– о временных ограничениях по доставке грузов в пункты назначения и их вывозу из пунктов отправления;

– о расстояниях между пунктами погрузки и разгрузки;

– об удельных временных затратах на перемещение груза на единицу расстояния на каждом участке маршрута;

– об удельных материальных затратах на перемещение единицы груза на единицу расстояния;

– о временных затратах на погрузку и разгрузку;

– о возможных временных задержках по ходу маршрута и другие.

2. *Определение схемы организации перевозок*. Определение производится на основании полученных заказов на доставку. Схемы подразделяются на схему «один к одному», схему «один ко многим» и «многие ко многим». Схема «один к одному» подразумевает, что от одного поставщика (из одного грузообразующего пункта) необходимо доставить груз только одному заказчику (в один грузопринимающий пункт). Под схемой «один ко многим» подразумевается, что доставка будет выполняться от одного поставщика и доставляться многим заказчикам. При организации перевозок по схеме «многие ко многим» товар доставляется от нескольких производителей или нескольких складов одного производителя нескольким заказчикам. Существует также схема «многие к одному» [5]. Автором [4] данная схема исключена из рассмотрения. Согласно проведенному им анализу публикаций, часто встречающиеся схемы доставки исчерпываются тремя ранее указанными схемами.

3. *Моделирование процесса перевозки*. В самом общем случае, когда при перевозке груза используется схема «многие ко многим», осуществляется решение транспортной задачи. Формальная постановка классической транспортной задачи следующая:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}\cdot x_{ij} \to \min_{x_{ij}\in\Omega}, \qquad (0.3)$$

$$\Omega: \quad \sum_{j=1}^{m} x_{ij} = a_i, \quad i = \overline{1,n}, \tag{0.4}$$

$$\sum_{i=1}^{n} x_{ij} = b_j, \quad j = \overline{1,m}, \tag{0.5}$$

$$x_{ij} \geq 0, \quad i = \overline{1,n}, \quad j = \overline{1,m}, \tag{0.6}$$

где *n* – количество поставщиков (грузоотправителей); *m* – количество потребителей (грузополучателей); a_i – ограничение по предложению (суммарное количество груза, поставляемое *i*-м поставщиком); b_j – ограничение по спросу (суммарная потребность в количестве груза у *j*-го получателя); c_{ij} – весовой коэффициент, определяющий вклад в формирование целевой функции единицы груза, перевозимого между *i*-м поставщиком и *j*-м получателем; x_{ij} – объем корреспонденции (количество транспортируемого груза) между *i*-м поставщиком и *j*-м получателем.

4. *Решение транспортной задачи.* Этап предполагает определение неизвестных x_{ij}, $i = \overline{1,n}$, $j = \overline{1,m}$, в математической модели (0.4) – (0,6), доставляющих минимум функции затрат на перевозку груза (0.3). Даже при малом числе поставщиков и потребителей $(n, m < 10)$ решение транспортной задачи методом прямого перебора всех возможных вариантов связано со значительными трудностями. При числе поставщиков и потребителей $(n, m \geq 10)$ оптимальное решение задачи методом прямого перебора проблематично даже с использованием современной вычислительной техники.

5. *Определение маршрутов перевозок*. На данном этапе определяется, по каким маршрутам – маятниковым (веерным) или развозным (круговым)– будет доставляться груз от поставщика к потребителям, закрепленным за ним, по результатам решения транспортной задачи (0.3) – (0.7). В случае, когда поставщика и потребителей связывает схема «один ко многим», решается задача

маршрутизации для поставщика относительно закрепленных за ним клиентов. Согласно [4], решение данной задачи сводится к решению задачи коммивояжера.

6. *Моделирование времени перевозок*. Моделируется время доставки товара каждому из потребителей. Для определения времени нахождения транспортного средства в наряде используется формула

$$T_н = \sum_{j}^{A} t_{nj} + \sum_{k}^{B} t_{грk} + \sum_{j}^{C} t_{pj} + \sum_{k}^{B} t_{xk} + \sum_{j}^{A} \eta_j + \sum_{i}^{C} \psi_i , \quad (0.8)$$

где t_{nj} – время погрузки у *j*-го поставщика, включая время ожидания погрузки; *A* – количество поставщиков, обеспечивающих загрузку транспортного средства, $A \le n$; $t_{грk}$ – время движения транспортного средства с грузом на *k*-м участке; *B* – количество неоднородных участков на маршруте; t_{xk} – время движения транспортного средства без груза на k-м участке; t_{pi} – время разгрузки у *i*-го потребителя, включая время ожидания разгрузки; *C* – количество потребителей, которым развозится груз, $C \le m$; η_j – случайная величина, учитывающая обеденные (технологические) перерывы *j*-го поставщика; ψ_i – случайная составляющая, учитывающая обеденные (технологические) перерывы *i*-го потребителя.

Для международной перевозки общее время нахождения автомобиля в рейсе (на маршруте) определяется по следующей формуле

$$T_o = \sum_{i=1}^{A} t_{i,i+1} + \sum_{j=1}^{B} \tau_j + \sum_{k=1}^{C} \Theta_k + \sum_{l=1}^{D} \varphi_l + \sum_{m=1}^{E} \psi_m + \sum_{n=1}^{F} \eta_n , \quad (0.9)$$

где $t_{i,i+1}$ – время движения между *i*-м и $(i+1)$-м пунктами маршрута; A – количество участков движения автомобиля на маршруте; τ_j – время оформления таможенных документов в *j*-м пункте; *B* –

количество пунктов таможенного оформления на маршруте; Θ_k – время погрузки, разгрузки и складирования в *k*-м пункте; *C* – количество пунктов погрузки-разгрузки; φ_l – случайная составляющая, отражающая увеличение времени рейса для проведения ремонтно-профилактических работ и других причин простоя транспортного средства; ψ_m – случайная составляющая, отражающая ограничения, связанные с ЕСТР; η_n – случайная составляющая, отражающая запреты на движения большегрузных автомобилей; D, E, F – число случаев простоя автомобиля с учетом последних трех указанных факторов соответственно.

Рассчитав время рейса, можно определить время прибытия к каждому из потребителей.

7. *Анализ выполнения требований и условий перевозок*. Выполняется проверка соответствия ожидаемых сроков доставки срокам, оговоренным с потребителем. В случае, если поставленные потребителем условия не выполняются, осуществляется корректировка маршрутов, грузоподъемности транспортного средства, расписания работы складов и так далее. В случае, когда корректировка приводит к изменению математической модели процесса перевозки, этапы 4 – 7 выполняются заново.

8. *Загрузка транспортного средства и непосредственная реализация перевозок*. Фактически, этап является критерием всего процесса перевозки. Нормальное завершение этапа свидетельствует о правильной организации транспортирования груза. Если выполнение этапа выявляет какие-либо нежелательные отклонения в запланированном процессе перевозки, то их оперативное устранение осуществляется не в рамках данного алгоритма. Все выявленные отклонения, как нежелательные, так и позитивные, должны быть учтены при моделировании будущих перевозок.

Недостатком приведенного алгоритма является его трудоемкость и ориентированность на небольшое количество поставщиков и потребителей [4, 6]. Кроме того, нерациональным является построение маршрутов без учета грузоподъемности автомобиля, осуществляющего поездку. В частности, для группы «поставщик – закрепленные потребители», возможно, придется решать задачу раз-

возки, а не задачу коммивояжера. Решение транспортной задачи с учетом грузоподъемности доступных транспортных средств может не совпадать с решением, полученным без учета грузоподъемности ТС. В статье [6] авторы предлагают следующие дополнения. 1) После формирования базы данных с информацией от транспортной сети предлагается локализовать склады и клиентов по территориальному признаку. 2) Процесс маршрутизации совместить с выбором транспортных средств соответствующей грузоподъемности.

Приведенная схема при планировании реальных грузовых автомобильных перевозок, как правило, имеет линейную структуру – все этапы выполняются последовательно один за другим. Но, в общем случае стремление учесть все факторы, влияющие на процесс транспортирования груза, все альтернативные маршруты и способы доставки приводит к нарушению линейности. При этом схема может включать как линейные, так и разветвлённые и циклические участки.

Для повышения эффективности процесса транспортных перевозок, в т.ч. и автомобильных перевозок, могут быть использованы научный подход с применением разработанных теоретически схем и моделей или эвристический подход, реализуемый экспертами компании, осуществляющей перевозку груза, на основании предыдущего опыта [6]. Кроме того, могут быть применены стандартные пакеты программ, предназначенные для оптимизации некоторых этапов процесса транспортных перевозок. Каждый из этих подходов или инструментов может быть определяющим или присутствовать наряду с другими. Исследование процесса транспортных перевозок ведется как за рубежом, так и в странах СНГ, в частности, в России и Украине. Согласно докладу министра России в 2008 году себестоимость автомобильных перевозок в России в полтора раза превышает их себестоимость в развитых зарубежных странах. В большинстве случаев затраты могут быть значительно уменьшены за счет оптимизации и автоматизации процесса доставки [7]. Согласно данным украинской конференции [8] в Украине на практике чаще всего используется второй из перечисленных подходов, т.е. планирование экспертами на основании наработанного опыта. Также применяются коммерческие пакеты программ, однако согласно тому же источнику оба подхода обеспечивают примерно одинаковую эффективность. Сопоставляя эти выводы с выводами, сделанными в источнике [7] по России, можно прийти к заключению, что коммер-

ческие пакеты программ применяются неэффективно, либо они не обеспечивают должной оптимальности организации процесса транспортных перевозок.

В практике организации транспортных перевозок наиболее широкое коммерческое применение нашли следующие программные пакеты:

1. Системы GPS-контроля. Данное программное обеспечение предназначено исключительно для контроля сотрудников и лиц, осуществляющих перевозки, и таким образом позволяющее исключить затраты, вызванные злоупотреблениями сотрудников. Это программное обеспечение не предназначено для оптимизации транспортного процесса.

2. Системы TMS-управления. Данное программное обеспечение представляет собой универсальный пакет, позволяющий выполнять следующие операции:

- управление заказами на перевозку;
- планирование и формирование маршрутов;
- обслуживание нестандартных транспортных событий;
- возможность взаимодействия с различного типа мобильными устройствами;
- расчет стоимости транспортных услуг;
- обслуживание договоров с внешними транспортными компаниями;
- задание пользователем прейскурантов транспортных услуг;
- задание пользователем алгоритмов расчета транспортных услуг;
- статистика и анализ данных по транспортной логистике.

3. Географические информационные системы (ГИС). Программное обеспечение данного типа позволяет строить маршруты в автоматическом режиме (с использованием встроенной в программу карты местности с дорогами) и определять время прохождения маршрутов. Последние версии усовершенствованных систем TMS-управления используют ГИС для построения маршрутов.

Приведенные программные средства позволяют решать задачи организации перевозки груза оптимальным образом только для транспортных сетей небольшой размерности. В случае транспортных сетей большой размерности существующее программное обеспечение, в т.ч. и приведенное, либо не позволяет получить какое-либо решение, либо не обеспечивает его оптимальность. В последнем случае квазиоптимальность решения может послужить причиной неоправданно высоких транспортных расходов.

Дополнительным недостатком существующего программного обеспечения является использование критериев оптимизации с ограниченной гибкостью, что значительно снижает их прикладное значение.

На основании изученной литературы можно сделать вывод, что организации транспортного процесса с точки зрения логистики является актуальной задачей. В свою очередь, системное решение задачи об эффективной организации автомобильных перевозок требует разработки методов, позволяющих эффективно решать ее подзадачи. Некоторым из подзадач в литературных источниках уделяется много внимания, другие – мало изучены. Методы решения многих подзадач были разработаны без учета системности.

Таким образом, изучение возможностей по улучшению организации транспортного процесса, безусловно, остаётся насущной проблемой. Успешное её решение может быть достигнуто только при комплексном рассмотрении всех этапов и составляющих процесса с привлечением дополнительного контингента учёных и исследователей.

РАЗДЕЛ 1

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ТРАНСПОРТНЫХ ЗАДАЧ

Среди всех этапов планирования транспортных перевозок этапы «моделирование процесса перевозок» и «решение транспортной задачи» остаются центральными. Объясняется это тем, что моделирование процесса перевозки с последующим определением оптимального распределения наличных объёмов груза у поставщиков между получателями позволяет не только в полной мере обеспечить их требования, но и сделать это с минимальными затратами. Другими словами, такой подход позволяет «на кончике пера» получить прибыль, которая теряется при эвристическом или интуитивном планировании перевозок.

Приведенная ранее классическая модель перевозок в «чистом» виде встречается довольно редко. Существует несколько разновидностей транспортной задачи [14], которые отличаются друг от друга математической моделью. Отличия могут касаться как вида целевой функции, так и вида ограничений или того и другого одновременно. Но в любом случае транспортная задача остаётся частным случаем общей задачи линейного программирования.

На рис. 1.1 приведена схема, отражающая классификацию наиболее часто встречающихся разновидностей транспортных задач. Как следует из схемы, все транспортные задачи по содержанию и по особенностям математической модели можно разделить на девять классов:

- перевозка грузов;
- распределение транспортных средств;
- выбор средств доставки груза;
- расстановка грузового флота;

Рис. 1.1 – Классификация транспортных задач по виду математической модели

- развозка грузов;
- закрытие предприятия;
- двухэтапная перевозка грузов;
- розыгрыш кубка;
- задача коммивояжера.

Наиболее известные транспортные задачи связаны с перевозкой грузов. Неслучайно данному классу задач принадлежит классическая транспортная задача (0.3) – (0.7). Данный класс не только наиболее известен, но и наиболее представителен. Он включает четыре типа задач:

- с непрерывной закрытой математической моделью;
- с непрерывной открытой математической моделью;
- с дискретной закрытой математической моделью;
- с дискретной открытой математической моделью.

Второй класс связан с выбором средств доставки груза и включает два типа задач:

- с простой математической моделью;
- с фиксированными доплатами.

Следующий класс транспортных задач связан с двухэтапной перевозкой груза. Он включает три типа задач:

- с простой математической моделью;
- с многомерной математической моделью;
- с многомерной математической моделью по запросам.

Каждый из оставшихся классов задач представлен одним типом математических моделей.

Все типы математических моделей, рассматриваемые в данной монографии, имеют названия, под которыми они чаще всего упоминаются в существующих литературных источниках [14 – 17].

Все типы задач будут подробно рассмотрены в текущем разделе. Для каждой задачи будут приведены её общие содержательная и математическая постановки, конкретный пример, цифровая модель в информационной среде *Microsoft Excel* и решение примера с помощью встроенного программного средства «Поиск решения».

Такое рассмотрение может послужить прекрасным справочником для построения математических моделей и методическим пособием для решения реальных производственных транспортных задач малой размерности. В данном случае размерность задач ограничивается вычислительными ресурсами компьютера, на котором инсталлирована среда *Microsoft Excel* и алгоритмическими особенностями программы «Поиск решения».

1.1. Транспортная задача планирования перевозок с непрерывной закрытой математической моделью

Содержательная постановка задачи

Общая постановка транспортной задачи планирования перевозок с непрерывной математической моделью состоит в определении оптимального плана перевозок некоторого однородного груза из m пунктов производства $A_1, A_2, \ldots, A_n$ в n пунктов потребления $B_1, B_2, \ldots, B_n$. Для каждого пункта отправления заданы объёмы производства $a_1, a_2, \ldots, a_m$. Для каждого пункта назначения заданы объёмы потребления $b_1, b_2, \ldots, b_n$. Требуется составить такой план перевозок, который полностью обеспечил бы всех потребителей при минимальных затратах на перевозку. При этом в качестве критерия оптимальности обычно берется либо минимальная стоимость перевозок всего груза, либо минимальное время его доставки, либо минимальный суммарный пробег грузового транспорта.

Считается, что общая потребность в грузе в пунктах назначения не превышает суммарных запасов груза в пунктах отправления. В противном случае удовлетворить потребности потребителей принципиально невозможно.

Математическая постановка задачи

Введём обозначения:

- c_{ij} – тарифы (время, расстояние) перевозки единицы груза из i–го пункта отправления в j–й пункт назначения;
- a_j – запасы груза в i–м пункте отправления;
- b_j – потребность в грузе в j–м пункте назначения;

- x_{ij} – количество ед. груза, перевозимого из *i*–го пункта отправления в *j*–й пункт назначения.

Тогда математическая постановка транспортной задачи о планировании перевозок имеет вид:

$$y = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \to \min_{x_{ij} \in \Omega}, \tag{1.1}$$

$$\Omega: \quad f_i = \sum_{j=1}^{n} x_{ij} \le a_i, \quad i = \overline{1,m}, \tag{1.2}$$

$$f_{m+j} = \sum_{i=1}^{m} x_{ij} = b_j, \quad j = \overline{1,n}, \tag{1.3}$$

$$x_{ij} \ge 0; \quad i = \overline{1,m}; \quad j = \overline{1,n}. \tag{1.4}$$

Здесь (1.1) – целевая функция, определяющая стоимость перевозок всего груза. Именно экстремальное (минимальное) значение этой функции необходимо найти в процессе решения задачи. Причем значения переменных x_{ij}, при которых целевая функция достигает своего минимума, должны принадлежать области допустимых решений Ω.

Выражения (1.2) – (1.4) определяют область допустимых решений Ω. Здесь система равенств (1.2) определяет суммарные объёмы перевозимого груза из каждого пункта отправления, система равенств (1.3) отражает суммарные объёмы груза от всех потребителей, поставляемые в каждый пункт назначения, а выражение (1.4) отсекает отрицательную область значений x_{ij}, в которую данные переменные не могут попадать по своему физическому смыслу.

Выражения (1.2) – (1.4) называют ограничениями задачи. Решение задачи (частный набор значений переменных x_{ij}) называют допустимым, если оно одновременно удовлетворяет всем ограничениям задачи. Решение задачи называют оптимальным, если оно

является допустимым и доставляет оптимум (в данном случае минимум) функции цели.

Функции $y,\ f_1, f_2, \dots, f_{m+n}$ являются непрерывными линейными функциями, заданными на неотрицательном октанте евклидова пространства $\mathbf{E}^n$. Данные функции имеют место, когда перевозимый груз является жидкостью, сыпучим материалом, мелкими заготовками, комплектующими или мелкой неупакованной продукцией. Такой груз характеризуется параметрами, которые представляют собой вес, погонные метры, квадратные метры, объем и т.п., но не штуки, упаковки, партии и тому подобное.

Если общая потребность в грузе в пунктах назначения равна суммарному запасу груза в пунктах отправления, т.е.

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j \ , \tag{1.5}$$

то модель такой транспортной задачи называется закрытой или сбалансированной, а сама задача – классической.

Условие (1.5) в явном виде не присутствует в ограничениях транспортной задачи. Оно косвенно учитывается системами ограничения (1.2) и (1.3). Причём для закрытой задачи система ограничений (1.2) трансформируется в систему равенств

$$f_i = \sum_{j=1}^{n} x_{ij} = a_i, \quad i = \overline{1,m} \ . \tag{1.6}$$

Пример транспортной задачи о планировании перевозок с непрерывной закрытой математической моделью

Пусть четыре предприятия некоторого экономического района для производства продукции используют три вида сырья. Потребности в сырье каждого из предприятий соответственно равны 120, 50, 190, и 110 *ед*. Сырье сосредоточено в трех местах его получения, а запасы соответственно равны 160, 140, 170 *ед*. На каждое из предприятий сырье может завозиться из любого пункта его получения. Тарифы перевозок являются известными величинами и задаются

матрицей $\mathbf{C} = \begin{bmatrix} 7 & 8 & 1 & 2 \\ 4 & 5 & 9 & 8 \\ 9 & 2 & 3 & 6 \end{bmatrix}$ (в *ус.ед.*). Необходимо составить такой план перевозок, при котором общая стоимость является минимальной.

Суммарная потребность в грузе в пунктах назначения по условию задачи равна суммарному запасу груза в пунктах отправления. Следовательно, данная задача является закрытой транспортной задачей.

Математическая модель транспортной задачи о планировании перевозок в условиях примера

Перед составлением математической модели проверяем условие (1.6), чтобы убедиться в том, что математическая модель задачи является закрытой. Действительно, общая потребность в грузе 470 *ед.* (120+ 50+190+110) равна суммарным запасам 470 ед. (160+140+170).

Математическая модель выше приведенной задачи с использованием принятых обозначений для задачи (1.1) – (1.4) имеет вид:

$$y = 7x_{11} + 8x_{12} + x_{13} + 2x_{14} + 4x_{21} + 5x_{22} + 9x_{23} + \\ + 8x_{24} + 9x_{31} + 2x_{32} + 3x_{33} + 6x_{34} \to \min_{x_{ij} \in \Omega},$$

$$\Omega: \quad f_1 = x_{11} + x_{12} + x_{13} + x_{14} = 160,$$

$$f_2 = x_{21} + x_{22} + x_{23} + x_{24} = 140,$$

$$f_3 = x_{31} + x_{32} + x_{33} + x_{34} = 170,$$

$$f_4 = x_{11} + x_{21} + x_{31} = 120,$$

$$f_5 = x_{12} + x_{22} + x_{32} = 50,$$

$$f_6 = x_{13} + x_{23} + x_{33} = 190,$$

$$f_7 = x_{14} + x_{24} + x_{34} = 110,$$

$$x_{ij} \geq 0; \quad i = \overline{1,3}; \quad j = \overline{1,4}.$$

Цифровая модель и решение задачи в информационной среде *Microsoft Excel*

Распределение и назначение ячеек электронной таблицы для транспортной задачи могут быть следующими:

- ячейки B5:E7 – для искомых переменных задачи x_{ij}, $i = \overline{1,3}; \; j = \overline{1,4}$
- ячейки B9:E9 – для функций f_{3+j}, $j = \overline{1,4}$, с загруженными формулами для подсчёта сумм $x_{1j} + x_{2j} + x_{3j}$, $j = \overline{1,4}$;
- ячейки B11:E11 – для потребностей в грузе b_j, $j = \overline{1,4}$, в пунктах назначения;
- ячейки G5:G7 – для функций f_i, $i = \overline{1,3}$, с загруженными формулами для подсчёта сумм $x_{i1} + x_{i2} + x_{i3} + x_{i4}$, $i = \overline{1,3}$;
- ячейки I5:I7 – для запасов груза a_i, $i = \overline{1,3}$, в пунктах отправления;
- ячейки B13:E15 – для элементов тарифной матрицы **C**;
- ячейки B17:E19 – для промежуточных результатов с загруженными формулами для вычисления произведения $c_{ij}x_{ij}$, $i = \overline{1,3}, \; j = \overline{1,4}$;
- ячейка G18 – для значения целевой функции *y* с загруженной формулой для подсчёта двойной суммы $\sum_{i=1}^{3}\sum_{j=1}^{4} c_{ij}x_{ij}$.

Все ячейки, задействованные в цифровой модели, должны быть отформатированы. Для них следует выбрать *числовой* формат. Число десятичных знаков для числового формата устанавливается исходя из характера единиц измерения моделируемых вели-

чин. Выбор должен обеспечивать достаточную точность вычисления. Так, для целевой ячейки, которая предназначена для величины, измеряемой в денежных единицах, должен устанавливаться формат с двумя десятичными знаками. В данном примере для всех ячеек условно принят такой же формат.

После загрузки в ячейки всех необходимых констант и формул для вычисления промежуточных и конечного результатов следует выполнить необходимые установки данных в диалоговом окне команды *Сервис/Поиск решения*. В условиях примера установки определяются следующим образом:

- для целевой ячейки – **$G$18**, равной **минимальному значению;**
- для ячеек с изменяемыми переменными – **$B$5:$E$7;**
- для ограничений: **$B$5:$E$7 >= 0;**

  $B$9:$E$9 = $B$11:$E$11;

  $G$5:$G$7 = $I$5:$I$7.

Вид диалогового окна с требуемыми установками показан на (рис. 1.2).

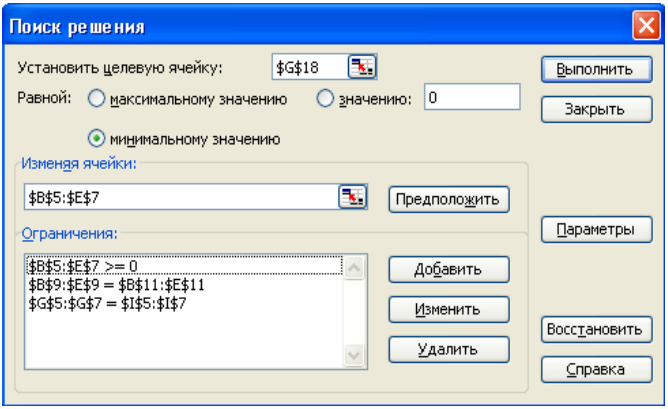
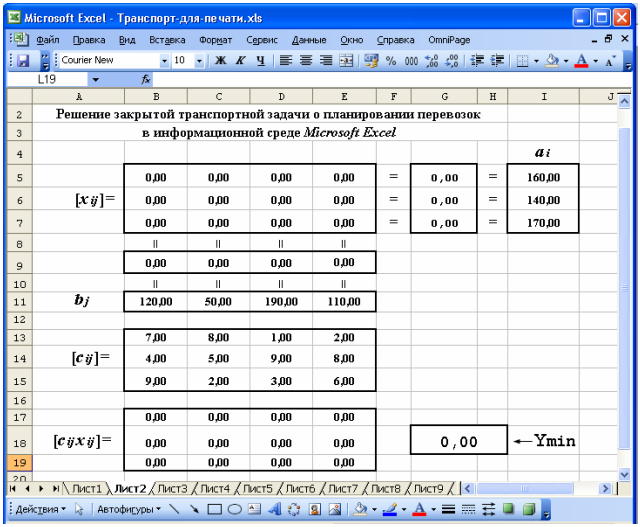


Рис. 1.2 – Диалоговое окно команды *Сервис/Поиск решения* после установки данных

На рис.1.3 показан вид экрана, который должен предшествовать выполнению команды *Сервис/Поиск решения*.

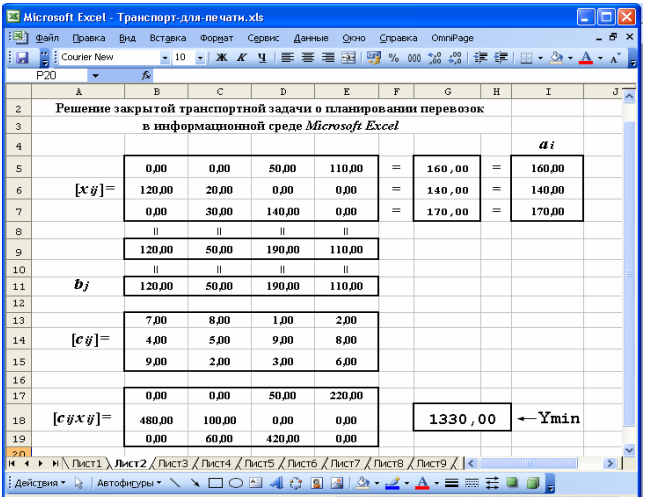


Решение закрытой транспортной задачи о планировании перевозок
в информационной среде *Microsoft Excel*

									a_i	
		0,00	0,00	0,00	0,00	=	0,00	=	160,00	
	$[x_{ij}]=$	0,00	0,00	0,00	0,00	=	0,00	=	140,00	
		0,00	0,00	0,00	0,00	=	0,00	=	170,00	
		=	=	=	=					
		0,00	0,00	0,00	0,00					
		=	=	=	=					
	b_j	120,00	50,00	190,00	110,00					
		7,00	8,00	1,00	2,00					
	$[c_{ij}]=$	4,00	5,00	9,00	8,00					
		9,00	2,00	3,00	6,00					
		0,00	0,00	0,00	0,00					
	$[c_{ij}x_{ij}]=$	0,00	0,00	0,00	0,00		0,00	←Ymin		
		0,00	0,00	0,00	0,00					

Рис. 1.3 – Экран с исходными установками

Вид экрана после выполнения команды *Сервис/Поиск решения* показан на рис. 1.4.



Решение закрытой транспортной задачи о планировании перевозок
в информационной среде *Microsoft Excel*

									a_i	
		0,00	0,00	50,00	110,00	=	160,00	=	160,00	
	$[x_{ij}]=$	120,00	20,00	0,00	0,00	=	140,00	=	140,00	
		0,00	30,00	140,00	0,00	=	170,00	=	170,00	
		=	=	=	=					
		120,00	50,00	190,00	110,00					
	b_j	120,00	50,00	190,00	110,00					
		7,00	8,00	1,00	2,00					
	$[c_{ij}]=$	4,00	5,00	9,00	8,00					
		9,00	2,00	3,00	6,00					
		0,00	0,00	50,00	220,00					
	$[c_{ij}x_{ij}]=$	480,00	100,00	0,00	0,00		1330,00	←Ymin		
		0,00	60,00	420,00	0,00					

Рис. 1.4 – Вид экрана с промежуточными и искомыми результатами

Оптимальным решением задачи является матрица значений переменных

$$\mathbf{X}^* = \begin{bmatrix} 0 & 0 & 50 & 110 \\ 120 & 20 & 0 & 0 \\ 0 & 30 & 140 & 0 \end{bmatrix},$$

которая обеспечивает минимум целевой функции

$$y^* = 1330 \text{ ус.ед.}$$

1.2. Транспортная задача планирования перевозок с непрерывной открытой математической моделью

Содержательная и математическая постановки задачи

Общая постановка транспортной задачи планирования перевозок с непрерывной открытой математической моделью совпадает с постановкой транспортной задачи с непрерывной открытой моделью. Однако для открытой модели должно выполняться условие превышения суммарных запасов груза в пунктах отправления над общей потребностью в грузе в пунктах назначения

$$\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j \ , \qquad (1.7)$$

Обозначения, используемые в математической модели транспортной задачи с непрерывной открытой моделью, совпадают с обозначениями и моделью закрытой транспортной задачи. Отличие между закрытой и открытой математическими моделями транспортной задачи заключается только в системе ограничений (1.2), которая в первом случае трансформируется в равенства, а во втором – остаётся без изменения.

В открытой транспортной задаче область допустимых значений для переменных значительно шире, поэтому целевая функция (при прочих равных условиях) достигает лучших значений. Следующий пример предоставляет возможность убедиться в этом утверждении.

Пример транспортной задачи о планировании перевозок с непрерывной открытой математической моделью

Пусть четыре предприятия города для производства продукции используют три вида сырья. Потребности в сырье каждого из предприятий соответственно равны 120, 50, 190, и 110 *ед*. Сырье сосредоточено в трех местах его получения, а запасы соответственно равны 170, 150, 180 *ед*. На каждое из предприятий сырье может завозиться из любого пункта его получения. Тарифы перевозок являются известными величинами и задаются матрицей $\mathbf{C} = \begin{bmatrix} 7 & 8 & 1 & 2 \\ 4 & 5 & 9 & 8 \\ 9 & 2 & 3 & 6 \end{bmatrix}$ (в *ус.ед.*). Необходимо составить такой план перевозок, при котором общая стоимость является минимальной.

Условие приведенного примера отличается от условия предыдущего примера только в данных, определяющих запасы сырья. Как следует из условия, величина общего запаса (170+150+180=500 *ед.*) превышают потребности предприятий (120+50+190+110=470 *ед.*). То есть математическая модель данной транспортной задачи является открытой.

Математическая модель транспортной задачи о планировании перевозок в условиях примера

Математическая модель выше приведенной задачи с использованием обозначений для задачи (1.1) – (1.4) имеет вид:

$$y = 7x_{11} + 8x_{12} + x_{13} + 2x_{14} + 4x_{21} + 5x_{22} + 9x_{23} + \\ + 8x_{24} + 9x_{31} + 2x_{32} + 3x_{33} + 6x_{34} \to \min_{x_{ij} \in \Omega},$$

$$\Omega: \quad f_1 = x_{11} + x_{12} + x_{13} + x_{14} \le 170,$$

$$f_2 = x_{21} + x_{22} + x_{23} + x_{24} \le 150,$$

$$f_3 = x_{31} + x_{32} + x_{33} + x_{34} \le 180,$$

$$f_4 = x_{11} + x_{21} + x_{31} = 120,$$

$$f_5 = x_{12} + x_{22} + x_{32} = 50,$$

$$f_6 = x_{13} + x_{23} + x_{33} = 190,$$

$$f_7 = x_{14} + x_{24} + x_{34} = 110,$$

$$x_{ij} \ge 0, \quad i = \overline{1,3}, \quad j = \overline{1,4}.$$

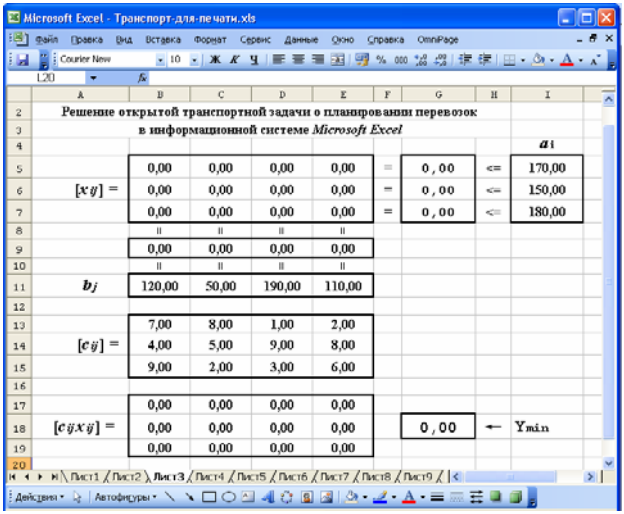
Цифровая модель и решение задачи в информационной среде *Microsoft Excel*

Распределение и назначение ячеек электронной таблицы для транспортной задачи могу быть такими же, как и в предыдущем примере. Однако установки в диалоговом окне команды должны быть следующими:

- для целевой ячейки – **$G$18**, равной **минимальному значению;**
- для ячеек с изменяемыми переменными – **$B$5:$E$7;**
- для ограничений: **$B$5:$E$7 >= 0;**
  $B$9:$E$9 = $B$11:$E$11;
  $G$5:$G$7 <= $I$5:$I$7 .

Здесь следует обратить внимание на последнее ограничение, которое, в отличие от аналогичного ограничения в ранее рассмотренном п.1.1, представляет собой систему неравенств.

На рис.1.5 показан вид экрана, который должен предшествовать выполнению команды *Сервис/Поиск решения*.



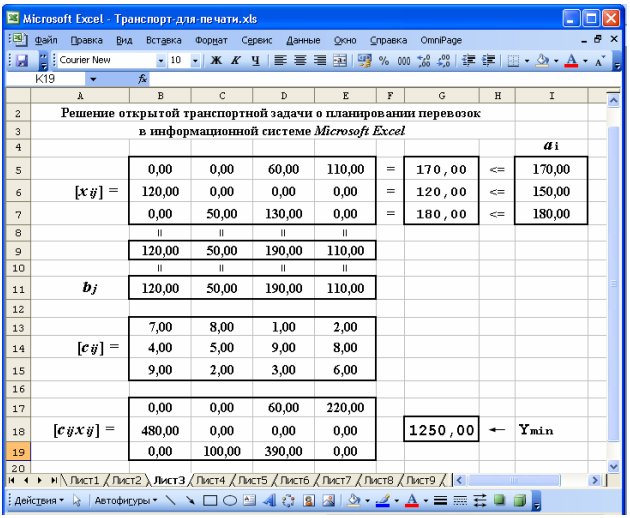
Microsoft Excel - Транспорт-для-печати.xls

Решение открытой транспортной задачи о планировании перевозок в информационной системе *Microsoft Excel*

									a_i
		0,00	0,00	0,00	0,00	=	0,00	<=	170,00
	$[x_{ij}]$ =	0,00	0,00	0,00	0,00	=	0,00	<=	150,00
		0,00	0,00	0,00	0,00	=	0,00	<=	180,00
		0,00	0,00	0,00	0,00				
	b_j	120,00	50,00	190,00	110,00				
		7,00	8,00	1,00	2,00				
	$[c_{ij}]$ =	4,00	5,00	9,00	8,00				
		9,00	2,00	3,00	6,00				
		0,00	0,00	0,00	0,00				
	$[c_{ij}x_{ij}]$ =	0,00	0,00	0,00	0,00		0,00	←	Y_{min}
		0,00	0,00	0,00	0,00				

Рис. 1.5 – Экран с исходными установками

Вид экрана после выполнения команды *Сервис/Поиск решения* показан на рис. 1.6.



Microsoft Excel - Транспорт-для-печати.xls

Решение открытой транспортной задачи о планировании перевозок в информационной системе *Microsoft Excel*

									a_i
		0,00	0,00	60,00	110,00	=	170,00	<=	170,00
	$[x_{ij}]$ =	120,00	0,00	0,00	0,00	=	120,00	<=	150,00
		0,00	50,00	130,00	0,00	=	180,00	<=	180,00
		120,00	50,00	190,00	110,00				
	b_j	120,00	50,00	190,00	110,00				
		7,00	8,00	1,00	2,00				
	$[c_{ij}]$ =	4,00	5,00	9,00	8,00				
		9,00	2,00	3,00	6,00				
		0,00	0,00	60,00	220,00				
	$[c_{ij}x_{ij}]$ =	480,00	0,00	0,00	0,00		1250,00	←	Y_{min}
		0,00	100,00	390,00	0,00				

Рис. 1.6 – Вид экрана с промежуточными и конечными результатами

Оптимальным решением задачи является матрица значений переменных

$$\mathbf{X}^* = \begin{bmatrix} 0 & 0 & 60 & 110 \\ 120 & 0 & 0 & 0 \\ 0 & 50 & 130 & 0 \end{bmatrix},$$

которая обеспечивает минимальное значение целевой функции

$$y^* = 1250 \text{ } \textit{ус.ед.}$$

Как видим, при прочих равных условиях решение открытой задачи приводит к лучшему результату. Затраты на перевозку по сравнению с предыдущей задачей сократились на 50 *ус.ед*.

1.3. Транспортная задача планирования перевозок с целочисленной закрытой математической моделью

Содержательная постановка задачи

Общая постановка транспортной задачи планирования перевозок с непрерывной математической моделью состоит в определении оптимального плана перевозок некоторого однородного груза из m пунктов производства $A_1, A_2, \ldots, A_m$ в n пунктов потребления $B_1, B_2, \ldots, B_n$. Для каждого пункта отправления заданы объёмы производства $a_1, a_2, \ldots, a_m$. Для каждого пункта назначения заданы объёмы потребления $b_1, b_2, \ldots, b_n$. Объёмы производства, потребления и перевозки измеряются только в целых единицах. Требуется составить такой план перевозок, который полностью обеспечил бы всех потребителей при минимальных затратах на перевозку. При этом, как и в транспортных задачах с непрерывной моделью, в качестве критерия оптимальности обычно берется либо минимальная стоимость перевозок всего груза, либо минимальное время его доставки, либо минимальный суммарный пробег грузового транспорта.

Считается, что общая потребность в грузе в пунктах назначения равна запасу груза в пунктах отправления.

Математическая постановка задачи

Введём обозначения, аналогичные обозначениям для непрерывной закрытой модели в п.1.1, а именно:

- c_{ij} – тарифы (время, расстояние) перевозки единицы груза из *i*–го пункта отправления в *j*–й пункт назначения;
- a_i – запасы груза в i–м пункте отправления;
- b_j – потребность в грузе в *j*–м пункте назначения;
- x_{ij} – количество ед. груза, перевозимого из i–го пункта отправления в *j*–й пункт назначения.

Тогда непрерывная закрытая математическая модель транспортной задачи о планировании перевозок будет иметь вид:

$$y = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \to \min_{x_{ij}\in\Omega}, \tag{1.8}$$

$$\Omega: \quad f_i = \sum_{j=1}^{n} x_{ij} = a_i, \quad i = \overline{1,m}, \tag{1.9}$$

$$f_{m+j} = \sum_{i=1}^{m} x_{ij} = b_j, \quad j = \overline{1,n}, \tag{1.10}$$

$$x_{ij} \ge 0; \quad i = \overline{1,m}; \quad j = \overline{1,n}, \tag{1.11}$$

$$x_{ij} = \mathrm{int}; \quad i = \overline{1,m}; \quad j = \overline{1,n}. \tag{1.12}$$

Закрытая целочисленная математическая модель транспортной задачи (1.8) – (1.12) отличается от ранее рассмотренной непрерывной закрытой математической модели (1.1), (1.3), (1.4) и (1.6) только наличием дополнительной системы ограничений на целочисленность переменных x_{ij}, т.е. наличием системы (1.12).

Здесь $y,\ f_1, f_2, \dots, f_{m+n}$ – дискретные линейные функции, заданные на неотрицательном октанте пространства целых чисел $\mathbf{R}^n$. Данные функции, как правило, имеют место, когда перевозимый груз представляет собой счетное множество крупных заготовок или

комплектующих, неделимых продуктов производства, упакованных сыпучих материалов и т.п. Объем такого груза характеризуется величинами, выражаемыми в штуках, упаковках, партиях и т.п., т.е величинами, которые не могут дробиться.

Пример транспортной задачи о планировании перевозок с непрерывной закрытой математической моделью

В качестве примера возьмем задачу из п.1.1, которая рассматривалась как пример задачи с непрерывной закрытой моделью. Однако добавим в условие задачи фразу о том, что сырьё может транспортироваться только в упаковках и что вес сырья в упаковке составляет 1 *ед*.

Сделанное добавление превращает ранее рассмотренную задачу с непрерывной закрытой математической моделью в задачу с целочисленной закрытой моделью.

Отметим также, что наложение условия целочисленности резко сокращает область допустимых решений, превращая его из непрерывного замкнутого множества точек в счётное ограниченное множество. Как правило, при прочих равных условиях такое превращение приводит к ухудшению результатов оптимизации.

Математическая и цифровая модели транспортной задачи о планировании перевозок в условиях примера

Математическая модель задачи с учётом сделанных добавлений будет незначительно отличаться от математической модели в условиях примера п.1.1. Для выполнения требования целочисленности переменных придётся ввести в математическую модель ограничение

$$x_{ij} = \text{int}; \quad i = \overline{1,3}; \quad j = \overline{1,4}\,.$$

Цифровая модель новой задачи останется прежней, но с той лишь разницей, что массивы ячеек B5:E7, B9:E9, B11:E11, G5:G7 и I5:I7 будут иметь *числовой* формат с количеством десятичных знаков, равным 0. Формат остальных ячеек следует оставить прежним, т.е. с количеством десятичных знаков, равным 2.

После загрузки в ячейки всех констант и формул для вычисления промежуточных и конечного результатов следует выполнить необходимые установки данных в диалоговом окне команды *Сервис/Поиск решения* (рис. 1.7).

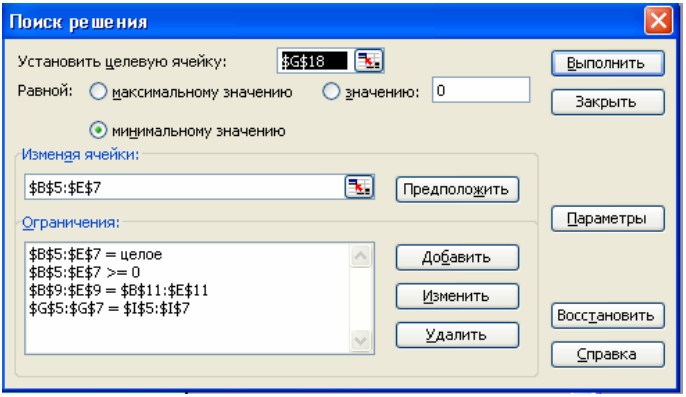


Рис. 1.7 – Диалоговое окно команды *Сервис/Поиск решения* после установки данных

Как видно из рис. 1.7, установки в диалоговом окне по сравнению с предыдущими примерами претерпели изменения. Теперь они следующие:

- для целевой ячейки – **$G$18**, равной **минимальному значению;**
- для ячеек с изменяемыми переменными – **$B$5:$E$7**;
- для ограничений: **$B$5:$E$7 >= 0**;

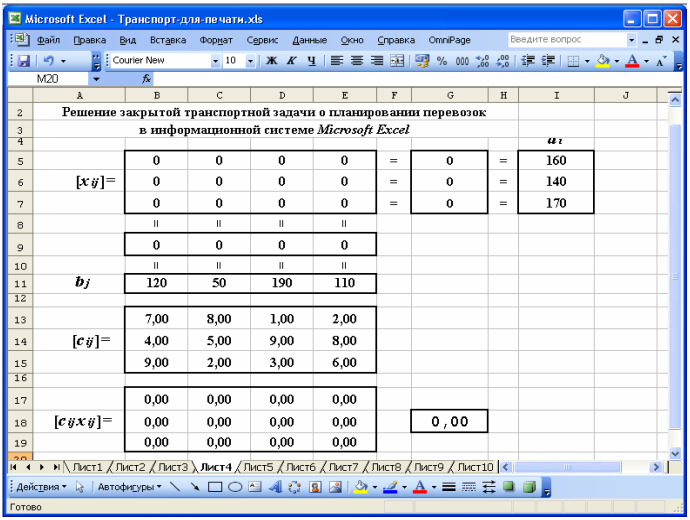
  $B$9:$E$9 = $B$11:$E$11;

  $G$5:$G$7 = $I$5:$I$7;.

  $B$5:$E$7 = целое ,

Установка последнего ограничения (**$B$5:$E$7 = целое**) при отработке команды *Сервис/Поиск решения* приводит к выполнению вычислительных операций только с целочисленными переменными.

На рис.1.8 показан вид экрана, который должен предшествовать выполнению команды *Сервис/Поиск решения*.

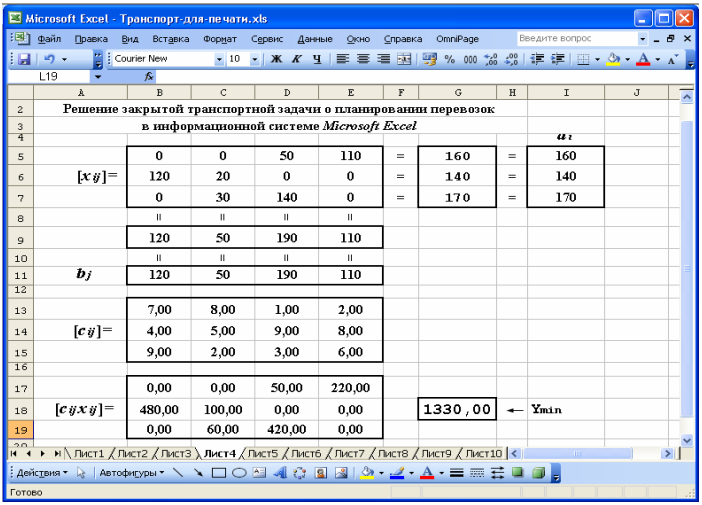


Решение закрытой транспортной задачи о планировании перевозок
в информационной системе *Microsoft Excel*

									a_i	
		0	0	0	0	=	0	=	160	
	$[x_{ij}]=$	0	0	0	0	=	0	=	140	
		0	0	0	0	=	0	=	170	
		=	=	=	=					
		0	0	0	0					
		=	=	=	=					
	b_j	120	50	190	110					
		7,00	8,00	1,00	2,00					
	$[c_{ij}]=$	4,00	5,00	9,00	8,00					
		9,00	2,00	3,00	6,00					
		0,00	0,00	0,00	0,00					
	$[c_{ij}x_{ij}]=$	0,00	0,00	0,00	0,00		0,00			
		0,00	0,00	0,00	0,00					

Рис. 1.8 – Экран с исходными установками

Вид экрана после выполнения команды *Сервис/Поиск решения* показан на рис. 1.9.



Решение закрытой транспортной задачи о планировании перевозок
в информационной системе *Microsoft Excel*

									a_i	
		0	0	50	110	=	160	=	160	
	$[x_{ij}]=$	120	20	0	0	=	140	=	140	
		0	30	140	0	=	170	=	170	
		=	=	=	=					
		120	50	190	110					
		=	=	=	=					
	b_j	120	50	190	110					
		7,00	8,00	1,00	2,00					
	$[c_{ij}]=$	4,00	5,00	9,00	8,00					
		9,00	2,00	3,00	6,00					
		0,00	0,00	50,00	220,00					
	$[c_{ij}x_{ij}]=$	480,00	100,00	0,00	0,00		1330,00	←	Y_{min}	
		0,00	60,00	420,00	0,00					

Рис. 1.9 – Вид экрана с промежуточными и искомыми результатами

Оптимальным решением задачи является матрица значений переменных

$$\mathbf{X}^* = \begin{bmatrix} 0 & 0 & 50 & 110 \\ 120 & 20 & 0 & 0 \\ 0 & 30 & 140 & 0 \end{bmatrix},$$

которая обеспечивает минимальное значение целевой функции

$$y^* = 1330 \text{ } \textit{ус.ед.}.$$

Сравнительный анализ полученного решения с решением задачи из п.1.1 показывает, что ввод ограничения целочисленности на переменные задачи не привел к ухудшению целевой функции (в обоих случаях она равна 1330 *ус. ед.*). Объясняется это тем, что в задаче из п.1.1 решение также оказалось целочисленным.

1.4. Транспортная задача планирования перевозок с целочисленной открытой математической моделью

Содержательная постановка задачи

Общая постановка транспортной задачи планирования перевозок с целочисленной открытой математической моделью совпадает с постановкой транспортной задачи с целочисленной закрытой моделью. Но, как и ранее, для открытой модели должно выполняться условие превышения суммарных запасов груза в пунктах отправления над общей потребностью в грузе в пунктах назначения.

Математическая постановка задачи

Математическая модель открытой транспортной задачи с целочисленными переменными будет иметь вид:

$$y = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \to \min_{x_{ij} \in \Omega}, \qquad (1.13)$$

$$\Omega: \quad f_i = \sum_{j=1}^{n} x_{ij} \le a_i, \quad i = \overline{1,m}\ , \tag{1.14}$$

$$f_{m+j} = \sum_{i=1}^{m} x_{ij} = b_j, \quad j = \overline{1,n}, \tag{1.15}$$

$$x_{ij} \ge 0; \quad i = \overline{1,m}; \quad j = \overline{1,n}, \tag{1.16}$$

$$x_{ij} = \text{int}; \quad i = \overline{1,m}; \quad j = \overline{1,n}. \tag{1.17}$$

Отличие данной модели от (1.8) – (1.12) наблюдается только в системе ограничений (1.14), где знак соотношения «=» сменился на «$\le$».

Пример транспортной задачи о планировании перевозок с непрерывной открытой математической моделью

В качестве примера оставим задачу задачи из п.1.2, но опять добавим в условие задачи фразу о том, что сырьё может транспортироваться только в упаковках и что вес сырья в упаковке составляет 1 *ед*. Такое добавление переведёт задачу из разряда непрерывных в разряд целочисленных.

Математическая модель транспортной задачи о планировании перевозок в условиях примера

Математическая модель выше приведенной задачи с использованием ранее принятых обозначений имеет вид:

$$y = 7x_{11} + 8x_{12} + x_{13} + 2x_{14} + 4x_{21} + 5x_{22} + 9x_{23} +$$
$$+ 8x_{24} + 9x_{31} + 2x_{32} + 3x_{33} + 6x_{34} \to \min_{x_{ij} \in \Omega},$$

$$\Omega: \quad f_1 = x_{11} + x_{12} + x_{13} + x_{14} \le 170,$$

$$f_2 = x_{21} + x_{22} + x_{23} + x_{24} \le 150,$$

$$f_3 = x_{31} + x_{32} + x_{33} + x_{34} \le 180,$$

$$f_4 = x_{11} + x_{21} + x_{31} = 120,$$

$$f_5 = x_{12} + x_{22} + x_{32} = 50,$$

$$f_6 = x_{13} + x_{23} + x_{33} = 190,$$

$$f_7 = x_{14} + x_{24} + x_{34} = 110,$$

$$x_{ij} \geq 0, \quad i = \overline{1,3}, \quad j = \overline{1,4},$$

$$x_{ij} = \text{int}, \quad i = \overline{1,3}, \quad j = \overline{1,4}.$$

Цифровая модель и решение задачи в информационной среде *Microsoft Excel*

Распределение и назначение ячеек электронной таблицы для транспортной задачи могут быть такими же, как и в предыдущем примере. Однако установки в диалоговом окне команды *Сервис/Поиск решения* должны быть следующими:

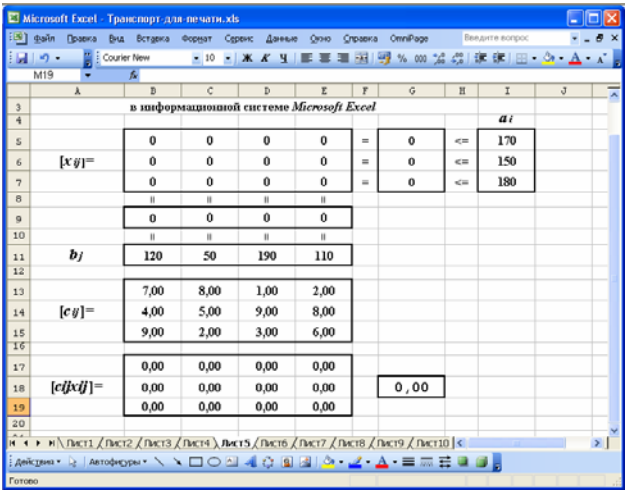
- для целевой ячейки – **$G$18**, равной **минимальному значению;**
- для ячеек с изменяемыми переменными – **$B$5:$E$7**;
- для ограничений: **$B$5:$E$7 >= 0**;
  $B$9:$E$9 = $B$11:$E$11;
  $G$5:$G$7 <= $I$5:$I$7;
  $B$5:$E$7 = целое.

Здесь следует обратить внимание на установку

$G$5:$G$7 <= $I$5:$I$7,

которая, в отличие от аналогичной установки в предыдущем примере, представляет собой систему неравенств.

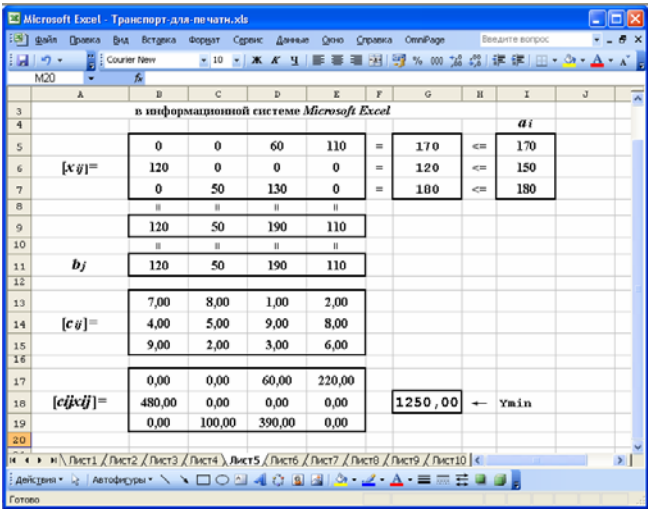
На рис.1.10 показан вид экрана, который должен предшествовать выполнению команды *Сервис/Поиск решения*.



3		в информационной системе *Microsoft Excel*							
4									*a i*
5		0	0	0	0	=	0	<=	170
6	[x ij]=	0	0	0	0	=	0	<=	150
7		0	0	0	0	=	0	<=	180
8		=	=	=	=				
9		0	0	0	0				
10		=	=	=	=				
11	*b j*	120	50	190	110				
12									
13		7,00	8,00	1,00	2,00				
14	[c ij]=	4,00	5,00	9,00	8,00				
15		9,00	2,00	3,00	6,00				
16									
17		0,00	0,00	0,00	0,00				
18	[cijxij]=	0,00	0,00	0,00	0,00		0,00		
19		0,00	0,00	0,00	0,00				
20									

Рис. 1.10 – Экран с исходными установками

Вид экрана после выполнения команды *Сервис/Поиск решения* показан на рис. 1.11.



3		в информационной системе *Microsoft Excel*							
4									*a i*
5		0	0	60	110	=	170	<=	170
6	[x ij]=	120	0	0	0	=	120	<=	150
7		0	50	130	0	=	180	<=	180
8		=	=	=	=				
9		120	50	190	110				
10		=	=	=	=				
11	*b j*	120	50	190	110				
12									
13		7,00	8,00	1,00	2,00				
14	[c ij]=	4,00	5,00	9,00	8,00				
15		9,00	2,00	3,00	6,00				
16									
17		0,00	0,00	60,00	220,00				
18	[cijxij]=	480,00	0,00	0,00	0,00		1250,00	←	Ymin
19		0,00	100,00	390,00	0,00				
20									

Рис. 1.11 – Вид экрана с промежуточными и конечными результатами

Оптимальным решением задачи является матрица значений переменных

$$\mathbf{X}^* = \begin{bmatrix} 0 & 0 & 60 & 110 \\ 120 & 0 & 0 & 0 \\ 0 & 50 & 130 & 0 \end{bmatrix},$$

которая обеспечивает минимальное значение целевой функции

$$y^* = 1250 \text{ уc.ед.}$$

Как видим, решение открытой задачи по сравнению с закрытой приводит к лучшему результату. Затраты на перевозку сократились на 50 *ус.ед*.

1.5. Задача о распределении транспортных средств с простой математической моделью

Содержательная постановка задачи

Пусть имеется n транспортных линий. По j–й линии необходимо выполнить b_j рейсов, $j=1,2,\ldots,n$. В наличии имеются транспортные единицы m типов. Резервы полезного времени транспортных единиц типа i составляют a_i, $i=1,2,\ldots,m$. На выполнение транспортной единицей типа i рейса на линии j требуется время t_{ij}, а затраты на рейс составляют c_{ij}. Требуется указать наиболее экономичную расстановку транспортных единиц по линиям.

Математическая постановка задачи

Обозначим через x_{ij} количество рейсов, которое транспортная единица i должна выполнить на линии j. Тогда математическая модель транспортной задачи о распределении транспортных средств будет иметь вид:

$$y = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \to \min_{x_{ij}\in\Omega}, \tag{1.18}$$

$$\Omega: \quad \sum_{j=1}^{n} t_{ij}x_{ij} \le a_i, \quad i = \overline{1,m}, \tag{1.19}$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = \overline{1,n}, \tag{1.20}$$

$$x_{ij} \ge 0; \quad i = \overline{1,m}; \quad j = \overline{1,n}; \tag{1.21}$$

$$x_{ij} = \text{int}; \quad i = \overline{1,m}; \quad j = \overline{1,n}. \tag{1.22}$$

Здесь целевая функция (1.18) соответствует суммарным затратам на выполнение всех рейсов по всем линиям. Система неравенств (1.19) ограничивают суммарные затраты времени транспортными единицами для каждого *i*–го типа. Система равенств (1.20) определяют общее количество рейсов на *j*-й линии. Выражения (1.21) и (1.22), обусловленные физической природой переменных, ограничивают пространство допустимых решений.

Пример транспортной задачи о распределении транспортных средств

Пусть имеется три транспортные линии, на которых необходимо выполнить соответственно 11, 7, 9 рейсов. В наличии имеются транспортные единицы трех типов. Резервы полезного времени транспортных единиц каждого типа соответственно составляют 100, 130, 250. Временные и стоимостные затраты на выполнение транспортной единицей определенного типа на каждой из трех линий задаются соответственно матрицами **T** и **C:**

$$\mathbf{T} = \begin{bmatrix} 4 & 3 & 3 \\ 2 & 2 & 3 \\ 2 & 2 & 2 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 5 & 5 & 4 \\ 3 & 3 & 5 \\ 4 & 3 & 4 \end{bmatrix}.$$

Требуется указать наиболее экономичную расстановку транспортных единиц по линиям.

Математическая модель транспортной задачи о распределении транспортных средств в условиях примера

Математическая модель задачи при использовании обозначений, принятых для общей модели задачи о распределении транспортных средств (1.18) – (1.22), будет иметь вид:

$$y = 5x_{11} + 5x_{12} + 4x_{13} + 3x_{21} + 3x_{22} + 5x_{23} +$$

$$+ 4x_{31} + 3x_{32} + 4x_{33} \to \min_{x_{ij} \in \Omega},$$

$$\Omega: \quad f_1 = 4x_{11} + 3x_{12} + 3x_{13} \le 100,$$

$$f_2 = 2x_{21} + 2x_{22} + 3x_{23} \le 130,$$

$$f_3 = 2x_{31} + 2x_{32} + 2x_{33} \le 250,$$

$$f_4 = x_{11} + x_{21} + x_{31} = 11,$$

$$f_5 = x_{12} + x_{22} + x_{32} = 7,$$

$$f_6 = x_{13} + x_{23} + x_{33} = 9,$$

$$x_{ij} \ge 0, \quad i = \overline{1,3}, \quad j = \overline{1,3},$$

$$x_{ij} = \text{int}; \quad i = \overline{1,3}, \quad j = \overline{1,3}.$$

Цифровая модель и решение задачи в информационной среде *Microsoft Excel*

Распределение и назначение ячеек электронной таблицы для задачи о распределении транспортных средств могут быть следующими:

- ячейки B5:D7 – для искомых переменных задачи x_{ij}, $i=\overline{1,3}, \quad j=\overline{1,3}$;
- ячейки B9:D9 – для функций f_{3+j}, $j=\overline{1,3}$, с загруженными формулами для подсчёта сумм $x_{1j}+x_{2j}+x_{3j}$, $j=\overline{1,3}$;
- ячейки B11:D11 – для необходимых количеств рейсов b_j, $j=\overline{1,3}$ на транспортных линиях;
- ячейки B13:D15 – для элементов тарифной матрицы **C**;
- ячейки B17: D19 – для промежуточных результатов с загруженными формулами для вычисления произведения $c_{ij}x_{ij}$, $i=\overline{1,3}$, $j=\overline{1,3}$;
- ячейки F5:H7 – для элементов временной матрицы **T**;
- ячейки J5:L7 – для промежуточных результатов с загруженными формулами, которые вычисляют произведения $t_{ij}x_{ij}$, $i=\overline{1,3}$, $j=\overline{1,3}$;
- ячейки N5:N7 – для функций f_i, $i=\overline{1,3}$, с загруженными формулами для подсчёта сумм $t_{i1}x_{i1}+t_{i2}x_{i2}+t_{i3}x_{i3}$, $i=\overline{1,3}$;
- ячейки P5:P7 – для резервов полезного времени каждого типа транспортного средства a_i, $i=\overline{1,3}$;
- ячейка F18 – для значения целевой функции *y* с загруженной формулой для подсчёта двойной суммы $\sum_{i=1}^{3}\sum_{j=1}^{3}c_{ij}x_{ij}$.

Все ячейки, задействованные в цифровой модели, должны иметь *числовой* формат с числом десятичных знаков, равным 2. Исключением является ячейка с целочисленными данными. Это ячейки B5:D7 (для искомых переменных задачи), B9:D9 (для сумм искомых переменных) и B11:D11 (для количеств рейсов на линии).

После загрузки в электронную таблицу всех необходимых констант и формул для вычисления промежуточных и конечного результатов следует выполнить необходимые установки данных в диалоговом окне команды *Сервис/Поиск решения*. В условиях примера установки определяются следующим образом:

- для целевой ячейки – **$F$18**, равной **минимальному значению;**
- для ячеек с изменяемыми переменными – **$B$5:$D$7;**
- для ограничений: **$B$5:$D$7 целое;**

  $B$5:$D$7 >= 0;

  $B$9:$D$9 = $B$11:$D$11;

  $N$5:$N$7 < = $P$5:$P$7.

На рис. 1.12 показан вид экрана, который должен предшествовать выполнению команды *Сервис/Поиск решения*.

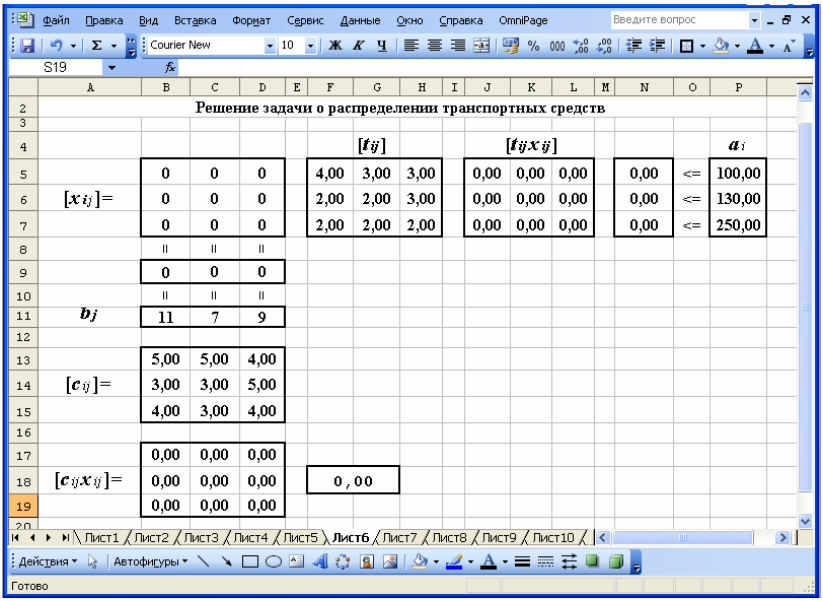


Рис. 1.12 – Экран с исходными установками

Вид экрана после выполнения команды *Сервис/Поиск решения* показан на рис. 1.13.

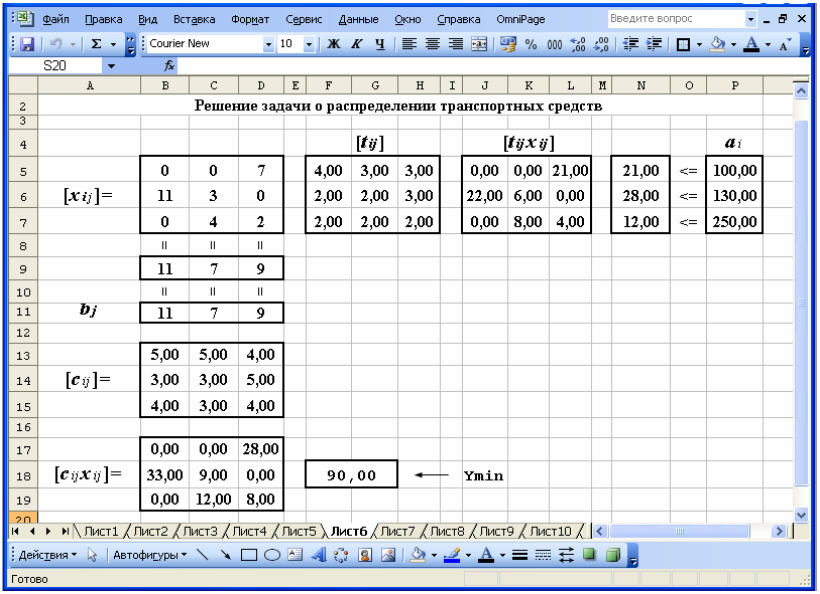


Рис. 1.13 – Вид экрана с промежуточными и конечными результатами

Оптимальным решением задачи является матрица значений переменных

$$\mathbf{X}^* = \begin{bmatrix} 0 & 0 & 7 \\ 11 & 3 & 0 \\ 0 & 4 & 2 \end{bmatrix},$$

которая обеспечивает минимальное значение целевой функции

$$y^* = 90 \text{ ус.ед.}$$

1.6. Задача о распределении транспортных средств с фиксированными доплатами

Содержательная постановка задачи

Пусть имеется *n* транспортных линий. По *j*–й линии необходимо выполнить b_j рейсов, $j=1,2,\ldots,n$. В наличии имеются транспортные единицы *m* типов. Резервы полезного времени транспортных единиц типа *i* составляют a_i, $i=1,2,\ldots,m$. На выполнение рейса одной транспортной единицей типа *i* по линии *j* требуется время t_{ij}, а затраты на рейс составляют c_{ij}. Кроме того, в отличие от ранее рассмотренной задачи о распределении транспортных средств, выпуск транспортной единицы типа *i* на линию *j* связан с подготовительными работами, требующими временных затрат t_{ij}^{+}. Это время не зависит от числа рейсов, которое должно выполнить транспортное средство. Денежные затраты на проведение этих подготовительных работ составляют c_{ij}^{+}. Требуется указать наиболее экономичную расстановку транспортных единиц по линиям.

Математическая постановка задачи

Как и в предыдущей задаче, обозначим через x_{ij} количество рейсов, которое транспортная единица *i* должна выполнить по линии *j*. Тогда математическая модель транспортной задачи о распределении транспортных средств с фиксированными доплатами примет вид:

$$y=\sum_{i=1}^{m}\sum_{j=1}^{n}s_{ij}(x_{ij})\to\min_{x_{ij}\in\Omega}, \tag{1.23}$$

$$\Omega:\quad \sum_{j=1}^{n}h_{ij}\left(x_{ij}\right)\le a_i,\quad i=\overline{1,m}, \tag{1.24}$$

$$\sum_{i=1}^{m}x_{ij}=b_j,\quad j=\overline{1,n}, \tag{1.25}$$

$$x_{ij} \geq 0; \quad i = \overline{1,m}; \quad j = \overline{1,n}; \tag{1.26}$$

$$x_{ij} = \text{int}; \quad i = \overline{1,m}; \quad j = \overline{1,n}. \tag{1.27}$$

где

$$s_{ij}(x_{ij}) = \begin{cases} 0, & x_{ij} = 0, \\ c_{ij}x_{ij} + c_{ij}^{+}, & x_{ij} > 0, \end{cases} \tag{1.28}$$

$$h_{ij}(x_{ij}) = \begin{cases} 0, & x_{ij} = 0, \\ t_{ij}x_{ij} + t_{ij}^{+}, & x_{ij} > 0. \end{cases} \tag{1.29}$$

Здесь целевая функция (1.23) соответствует суммарным затратам на выполнение всех рейсов по всем линиям с учётом фиксированных доплат. Система неравенств (1.24) в соответствии с условием примера ограничивает суммарные затраты времени транспортными единицами для каждого i–го типа также с учётом фиксированных доплат – затраты не должны превосходить резерва полезного времени. Система равенств (1.25) моделирует требования на общее количество рейсов на каждой линии. Системы (1.26) и (1.27) ограничивают пространство допустимых решений, обусловленное физической природой переменных.

Пример транспортной задачи о распределении транспортных средств

Пусть имеется три транспортные линии, на которых необходимо выполнить соответственно 11, 7, 9 рейсов. В наличии имеются транспортные единицы трех типов. Резервы полезного времени транспортных единиц каждого типа соответственно составляют 100, 130, 250. Временные и стоимостные затраты на выполнение транспортной единицей каждого типа на каждой из трех линий задаются соответственно матрицами: $\mathbf{T} = \begin{bmatrix} 4 & 3 & 3 \\ 2 & 2 & 3 \\ 2 & 2 & 2 \end{bmatrix}$, $\mathbf{C} = \begin{bmatrix} 5 & 5 & 4 \\ 3 & 3 & 5 \\ 4 & 3 & 4 \end{bmatrix}$.

Время t_{ij}^{+}, связанное с подготовительными работами на выпуск транспортной единицы типа i на линию j, и денежные затраты c_{ij} на проведение этих подготовительных работ задаются соответственно матрицами: $\mathrm{T}^{+} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 3 \\ 1 & 3 & 2 \end{bmatrix}$, $\mathbf{C}^{+} = \begin{bmatrix} 1 & 2 & 2 \\ 1 & 1 & 2 \\ 2 & 1 & 1 \end{bmatrix}$. Требуется указать наиболее экономичную расстановку транспортных единиц по линиям.

Математическая модель транспортной задачи о распределении транспортных средств в условиях примера

Математическая модель задачи при использовании обозначений, принятых для общей математической модели задачи о распределении транспортных средств (1.23) – (1.29), будет иметь вид:

$$y = s_{11} + s_{12} + s_{13} + s_{21} + s_{22} + s_{23} + s_{31} + s_{32} + s_{33} \to \min_{x_{ij} \in \mathbf{\Omega}}$$

$$\mathbf{\Omega}: \quad f_1 = h_{11} + h_{12} + h_{13} \le 100,$$

$$f_2 = h_{21} + h_{22} + h_{23} \le 130,$$

$$f_3 = h_{31} + h_{32} + h_{33} \le 250,$$

$$f_4 = x_{11} + x_{21} + x_{31} = 11,$$

$$f_5 = x_{12} + x_{22} + x_{32} = 7,$$

$$f_6 = x_{13} + x_{23} + x_{33} = 9,$$

$$x_{ij} \ge 0, \quad i = \overline{1,3}, \quad j = \overline{1,3},$$

$$x_{ij} = \text{int}, \quad i = \overline{1,3}, \quad j = \overline{1,3}.$$

Здесь

$$s_{11} = \begin{cases} 0, & x_{11} = 0, \\ 5x_{11} + 1, & x_{11} > 0, \end{cases} \qquad s_{12} = \begin{cases} 0, & x_{12} = 0, \\ 5x_{12} + 2, & x_{12} > 0, \end{cases}$$

$$s_{13} = \begin{cases} 0, & x_{13} = 0, \\ 4x_{13} + 2, & x_{13} > 0, \end{cases} \qquad s_{21} = \begin{cases} 0, & x_{21} = 0, \\ 3x_{21} + 1, & x_{21} > 0, \end{cases}$$

$$s_{22} = \begin{cases} 0, & x_{22} = 0, \\ 3x_{22} + 1, & x_{22} > 0, \end{cases} \qquad s_{23} = \begin{cases} 0, & x_{23} = 0, \\ 5x_{23} + 2, & x_{23} > 0, \end{cases}$$

$$s_{31} = \begin{cases} 0, & x_{31} = 0, \\ 4x_{31} + 2, & x_{31} > 0, \end{cases} \qquad s_{32} = \begin{cases} 0, & x_{32} = 0, \\ 3x_{32} + 1, & x_{32} > 0, \end{cases}$$

$$s_{33} = \begin{cases} 0, & x_{33} = 0, \\ 4x_{33} + 1, & x_{33} > 0, \end{cases} \qquad h_{11} = \begin{cases} 0, & x_{11} = 0, \\ 4x_{11} + 1, & x_{11} > 0, \end{cases}$$

$$h_{12} = \begin{cases} 0, & x_{12} = 0, \\ 3x_{12} + 2, & x_{12} > 0, \end{cases} \qquad h_{13} = \begin{cases} 0, & x_{13} = 0, \\ 3x_{13} + 1, & x_{13} > 0, \end{cases}$$

$$h_{21} = \begin{cases} 0, & x_{21} = 0, \\ 2x_{21} + 2, & x_{21} > 0, \end{cases} \qquad h_{22} = \begin{cases} 0, & x_{22} = 0, \\ 3x_{22} + 1, & x_{22} > 0, \end{cases}$$

$$h_{23} = \begin{cases} 0, & x_{23} = 0, \\ 3x_{23} + 3, & x_{23} > 0, \end{cases} \qquad h_{31} = \begin{cases} 0, & x_{31} = 0, \\ 2x_{31} + 1, & x_{31} > 0, \end{cases}$$

$$h_{32} = \begin{cases} 0, & x_{32} = 0, \\ 2x_{32} + 3, & x_{32} > 0, \end{cases} \qquad h_{33} = \begin{cases} 0, & x_{33} = 0, \\ 2x_{33} + 2, & x_{33} > 0. \end{cases}$$

Цифровая модель и решение задачи в информационной среде *Microsoft Excel*

Распределение и назначение ячеек электронной таблицы для задачи о распределении транспортных средств с фиксированными доплатами могут быть следующими:

- ячейки B5:D7 – для искомых переменных задачи x_{ij}, $i = \overline{1,3},\ \ j = \overline{1,3}$;
- ячейки B9:D9 – для функций $f_{3+j},\ \ j = \overline{1,3}$, с загруженными формулами для подсчёта сумм $x_{1j} + x_{2j} + x_{3j},\ \ j = \overline{1,3}$;
- ячейки B11:D11 – для необходимых количеств рейсов $b_j,\ \ j = \overline{1,3}$, на транспортных линиях, заданных условием примера;
- ячейки B13:D15 – для элементов тарифной матрицы **C**;
- ячейки F13:H15 – для элементов тарифной матрицы $\mathbf{C}^+$ с фиксированными доплатами на подготовку транспортных средств;
- ячейки F17:H19 – для промежуточных результатов с загруженными формулами для вычисления функций $s_{ij},\ \ i = \overline{1,3},\ \ j = \overline{1,3}$;
- ячейки F5:H7 – для элементов временной матрицы **T;**
- ячейки J5:L7 – для элементов временной матрицы $\mathbf{T}^+$ с фиксированными временными затратами на подготовку транспортных средств**;**
- ячейки J9:L11 – для промежуточных результатов с загруженными формулами, которые вычисляют функции $h_{ij},\ \ i = \overline{1,3},\ j = \overline{1,3}$;

- ячейки N9:N11 – для функций f_i, $i=\overline{1,3}$, с загруженными формулами для подсчёта сумм $h_{i1}+h_{i2}+h_{i3}$, $i=\overline{1,3}$;

- ячейки P5:P7 – для резервов полезного времени каждого типа транспортного средства a_i, $i=\overline{1,3}$, заданных условием примера;

- ячейка J18 – для значения целевой функции *y* с загруженной формулой для подсчёта двойной суммы $\sum_{i=1}^{3}\sum_{j=1}^{3} s_{ij}$.

Все ячейки, задействованные в цифровой модели, должны иметь *числовой* формат с числом десятичных знаков, равным 2. Исключением являются ячейки с целочисленными данными. Это ячейки B5:D7 (для искомых переменных задачи), B9:D9 (для сумм искомых переменных) и B11:D11 (для количеств рейсов на линии).

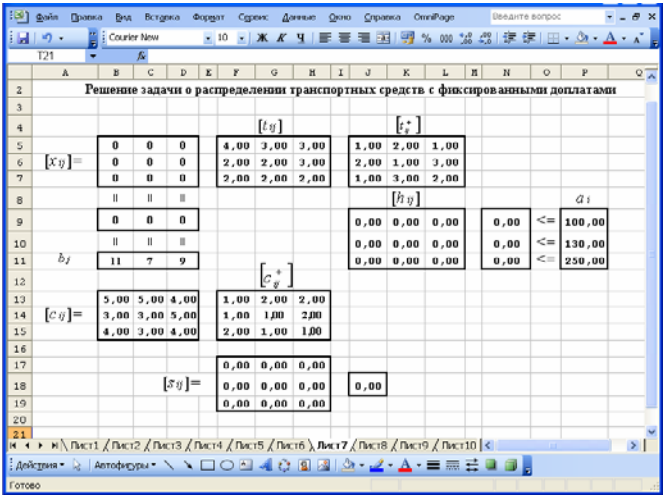
После загрузки в электронную таблицу всех необходимых констант и формул для вычисления промежуточных и конечного результатов следует выполнить необходимые установки данных в диалоговом окне команды *Сервис/Поиск решения*. В условиях примера установки определяются следующим образом:

- для целевой ячейки – **$J$18**, равной **минимальному значению;**

- для ячеек с изменяемыми переменными – **$B$5:$D$7;**

- для ограничений: **$B$5:$D$7 целое;**
  $B$5:$D$7 >= 0;
  $N$9:$N$11 <= $P$9:$P$11;
  $B$9:$D$9 = $B$11:$D$11.

На рис.1.14 показан вид экрана, который должен предшествовать выполнению команды *Сервис/Поиск решения*.

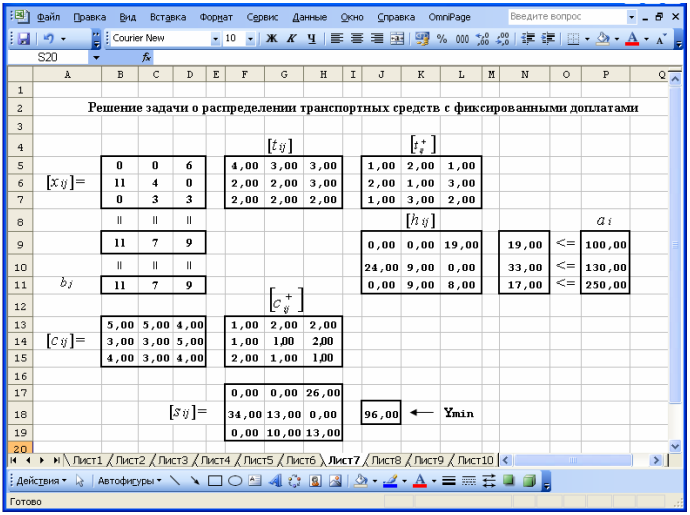


Решение задачи о распределении транспортных средств с фиксированными доплатами

							$[t_{ij}]$					$[t^+_{ij}]$						
		0	0	0		4,00	3,00	3,00		1,00	2,00	1,00						
$[x_{ij}]=$		0	0	0		2,00	2,00	3,00		2,00	1,00	3,00						
		0	0	0		2,00	2,00	2,00		1,00	3,00	2,00						
		II	II	II							$[h_{ij}]$						a_i	
		0	0	0						0,00	0,00	0,00		0,00		<=	100,00	
		II	II	II						0,00	0,00	0,00		0,00		<=	130,00	
b_j		11	7	9						0,00	0,00	0,00		0,00		<=	250,00	
							$[c^+_{ij}]$											
		5,00	5,00	4,00		1,00	2,00	2,00										
$[c_{ij}]=$		3,00	3,00	5,00		1,00	1,00	2,00										
		4,00	3,00	4,00		2,00	1,00	1,00										
						0,00	0,00	0,00										
	$[s_{ij}]=$					0,00	0,00	0,00		0,00								
						0,00	0,00	0,00										

Рис. 1.14 – Экран с исходными установками

Вид экрана после выполнения команды *Сервис/Поиск решения* показан на рис. 1.15.



Решение задачи о распределении транспортных средств с фиксированными доплатами

							$[t_{ij}]$					$[t^+_{ij}]$						
		0	0	6		4,00	3,00	3,00		1,00	2,00	1,00						
$[x_{ij}]=$		11	4	0		2,00	2,00	3,00		2,00	1,00	3,00						
		0	3	3		2,00	2,00	2,00		1,00	3,00	2,00						
		II	II	II							$[h_{ij}]$						a_i	
		11	7	9						0,00	0,00	19,00		19,00		<=	100,00	
		II	II	II						24,00	9,00	0,00		33,00		<=	130,00	
b_j		11	7	9						0,00	9,00	8,00		17,00		<=	250,00	
							$[c^+_{ij}]$											
		5,00	5,00	4,00		1,00	2,00	2,00										
$[c_{ij}]=$		3,00	3,00	5,00		1,00	1,00	2,00										
		4,00	3,00	4,00		2,00	1,00	1,00										
						0,00	0,00	26,00										
	$[s_{ij}]=$					34,00	13,00	0,00		96,00	←	Ymin						
						0,00	10,00	13,00										

Рис. 1.15 – Вид экрана с промежуточными и конечными результатами

Оптимальным решением задачи является матрица значений переменных

$$\mathbf{X}^* = \begin{bmatrix} 0 & 0 & 6 \\ 11 & 4 & 0 \\ 0 & 3 & 3 \end{bmatrix},$$

которая обеспечивает минимальное значение целевой функции

$$y^* = 96 \text{ уc.ед.}$$

1.7. Транспортная распределительная задача о расстановке флота

Содержательная постановка задачи

Пусть через $j=1,2,\ldots,n$ обозначены грузообразующие пункты с объемами груза в них a_j. Имеется m средств доставки груза (видов транспорта). Грузоподъемность i–го средства доставки составляет p_i, а наличный его парк равен bi, $i=1,2,\ldots,m$. Грузы подлежат доставке в один центральный пункт (склад). Затраты при осуществлении одной единицей i–го средства доставки от j–го пункта до склада равны c_{ij}. Требуется составить наиболее экономичный план доставки.

Математическая постановка задачи

Обозначим через x_{ij} количество средств доставки i–го типа, отправляющегося из j–го пункта. Тогда математическая модель распределительной транспортной распределительной задачи о расстановке флота имеет вид:

$$y = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \to \min_{x_{ij} \in \Omega}, \qquad (1.30)$$

$$\Omega: \quad \sum_{i=1}^{m} p_i x_{ij} \geq a_j, \quad j = \overline{1,n} \ , \tag{1.31}$$

$$\sum_{j=1}^{n} x_{ij} = b_i, \quad i = \overline{1,m}, \tag{1.32}$$

$$x_{ij} \geq 0, \quad i = \overline{1,m}, \quad j = \overline{1,n}, \tag{1.33}$$

$$x_{ij} = \text{int}, \quad i = \overline{1,m}, \quad j = \overline{1,n}. \tag{1.34}$$

Здесь целевая функция (1.30) определяет суммарные затраты на доставку груза на центральный склад. Выражения (1.31) указывают на необходимость вывоза всего груза с грузообразующих пунктов. Ограничение (1.32) указывает на то, что количество используемых средств доставки не должно превышать их наличный парк. Выражения (1.33) и (1.34), обусловлены физической природой переменных, ограничивают пространство допустимых решений.

Пример транспортной распределительной задачи о расстановке флота

На центральный склад груз поступает от трех предприятий соответственно в объемах 70, 120, 90 и 150 единиц. Наличный парк транспортных средств состоит из машин трех типов в количествах 50, 35 и 20 и грузоподъемности 5, 12 и 20 ед. соответственно. Затраты при осуществлении доставки груза одной единицей каждой из трех типов машин от каждого из трех предприятий заданы матрицей $\mathbf{C} = \begin{bmatrix} 5 & 5 & 4 & 6 \\ 3 & 3 & 5 & 2 \\ 4 & 3 & 4 & 5 \end{bmatrix}$. Требуется составить наиболее экономичный план выделения транспортных средств для предприятий, обеспечивающих поставку груза в указанных объемах.

Математическая модель транспортной распределительной задачи о расстановке флота в условиях примера

Математическая модель задачи при использовании обозначений, принятых для общей модели задачи о расстановке флота (1.30) – (1.34), будет иметь вид:

$$y = 5x_{11} + 5x_{12} + 4x_{13} + 6x_{14} + 3x_{21} + 3x_{22} + 5x_{23} + \\ + 2x_{24} + 4x_{31} + 3x_{32} + 4x_{33} + 5x_{34} \to \min_{x_{ij} \in \Omega},$$

$$\Omega: \quad f_1 = 5x_{11} + 12x_{21} + 20x_{31} \geq 70,$$

$$f_2 = 5x_{12} + 12x_{22} + 20x_{32} \geq 120,$$

$$f_3 = 5x_{13} + 12x_{23} + 20x_{33} \geq 90,$$

$$f_4 = 5x_{14} + 12x_{24} + 20x_{34} \geq 150,$$

$$f_5 = x_{11} + x_{12} + x_{13} + x_{14} = 50,$$

$$f_6 = x_{21} + x_{22} + x_{23} + x_{24} = 35,$$

$$f_7 = x_{31} + x_{32} + x_{33} + x_{34} = 20,$$

$$x_{ij} \geq 0, \quad i = \overline{1,3}, \quad j = \overline{1,4},$$

$$x_{ij} = \text{int}, \quad i = \overline{1,3}, \quad j = \overline{1,4}.$$

Цифровая модель и решение задачи в информационной среде *Microsoft Excel*

Распределение и назначение ячеек электронной таблицы для транспортной распределительной задачи о расстановке флота в условиях примера могут быть следующими:

- ячейки B5:E7 – для искомых переменных задачи x_{ij}, $i = \overline{1,3}, \quad j = \overline{1,4}$;
- ячейки B9:E11 – для элементов тарифной матрицы **C**;

- ячейки B13:E15 – для промежуточных результатов с загруженными формулами для вычисления произведения $c_{ij}x_{ij}$, $i=\overline{1,3}$, $j=\overline{1,4}$;
- ячейки G5:G7 – для указания грузоподъёмности транспортных средств p_i, $i=\overline{1,3}$;
- ячейки I5:L7 – для промежуточных результатов с загруженными формулами, которые вычисляют произведения $p_i x_{ij}$, $i=\overline{1,3}$, $j=\overline{1,4}$;
- ячейки I9:L9 – для функций f_j, $j=\overline{1,4}$, с загруженными формулами для подсчёта сумм $p_1x_{1j}+p_2x_{2j}+p_3x_{3j}$;
- ячейки I11:L11 – для указания объёма груза в пунктах отправления a_j, $j=\overline{1,4}$;
- ячейки N5:N7 – для функций f_{4+i}, $i=\overline{1,3}$, с загруженными формулами для подсчёта сумм $x_{i1}+x_{i2}+x_{i3}$;
- ячейки P5:P7 – для указания наличного количества транспортных средств b_i, $i=\overline{1,3}$;
- ячейка G14 – для значения целевой функции *y* с загруженной формулой для подсчёта двойной суммы $\sum_{i=1}^{3}\sum_{j=1}^{4}c_{ij}x_{ij}$.

Все ячейки, задействованные в цифровой модели, должны иметь *числовой* формат с числом десятичных знаков, равным 2. Исключение составляют ячейки с целочисленными данными. Это ячейки B5:E7 (для искомых переменных задачи), N5:N7 (для промежуточных результатов – сумм искомых переменных) и P5:P7 (для наличного количества транспортных средств).

После загрузки в электронную таблицу всех необходимых констант и формул для вычисления промежуточных и конечного результатов следует выполнить необходимые установки данных в диалоговом окне команды *Сервис/Поиск решения*. В условиях примера установки определяются следующим образом:

- для целевой ячейки – **$G$14**, равной **минимальному значению;**
- для ячеек с изменяемыми переменными – **$B$5:$E$7;**
- для ограничений: **$B$5:$E$7 >= 0**

  $B$5:$E$7 = целое

  $I$9: $L$9 >= $I$11:$L$11

  $N$5:$N$7 <= $P$5: $P$7

На рис.1.16 показан вид экрана, который должен предшествовать выполнению команды *Сервис/Поиск решения*.

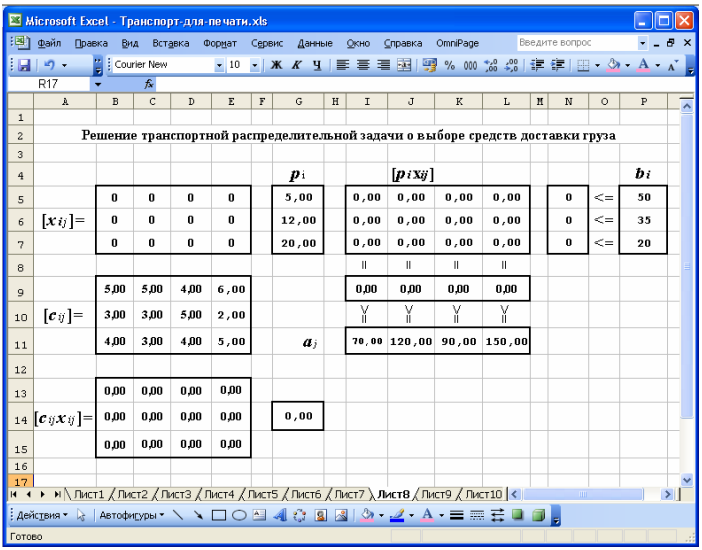
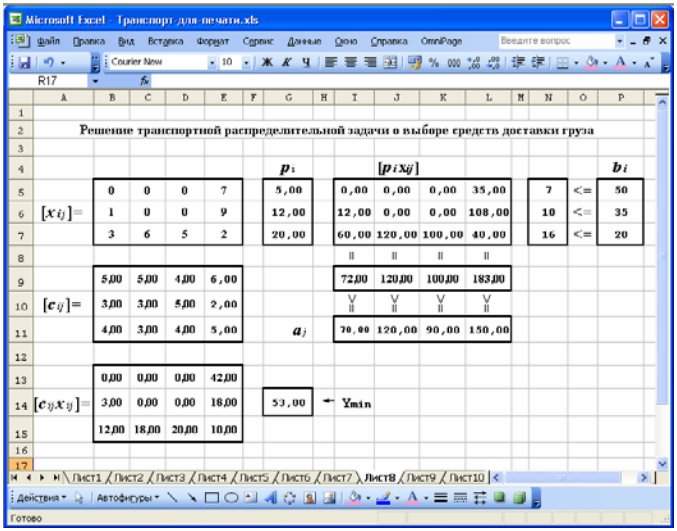


Рис. 1.16 – Экран с исходными установками

Вид экрана после выполнения команды *Сервис/Поиск решения* показан на рис. 1.17.



Microsoft Excel - Транспорт для печати.xls

Решение транспортной распределительной задачи в выборе средств доставки груза

						p_i		$[p_i x_{ij}]$						b_i
		0	0	0	7	5,00		0,00	0,00	0,00	35,00	7	<=	50
	$[x_{ij}]=$	1	0	0	9	12,00		12,00	0,00	0,00	108,00	10	<=	35
		3	6	5	2	20,00		60,00	120,00	100,00	40,00	16	<=	20
								=	=	=	=			
		5,00	5,00	4,00	6,00			72,00	120,00	100,00	183,00			
	$[c_{ij}]=$	3,00	3,00	5,00	2,00			>=	>=	>=	>=			
		4,00	3,00	4,00	5,00	a_j		70,00	120,00	90,00	150,00			
		0,00	0,00	0,00	42,00									
	$[c_{ij}x_{ij}]=$	3,00	0,00	0,00	18,00	53,00	←	Ymin						
		12,00	18,00	20,00	10,00									

Рис. 1.17 – Вид экрана с промежуточными и конечными результатами

Оптимальным решением задачи является матрица значений переменных

$$\mathbf{X}^* = \begin{bmatrix} 0 & 0 & 0 & 7 \\ 1 & 0 & 0 & 9 \\ 3 & 6 & 5 & 2 \end{bmatrix},$$

которая обеспечивает минимальное значение целевой функции

$$y^* = 53 \text{ ус.ед.}$$

1.8. Транспортная задача о выборе средств доставки груза

Содержательная постановка задачи

Пусть грузовой флот имеет в своем составе суда n типов. Количество судов типа j равно q_j, а затраты при использовании одного судна типа j в планируемом периоде составляет c_j, $j=1,2,\ldots,n$. Ка-

ждое судно обладает грузовыми емкостями m типов (трюмы, палубы, танки и т.п.). Грузоподъемность емкости i на судне типа j равна d_{ij}, $i=1,2,\ldots,m$. Перевозке подлежат p видов груза. Груз вида k имеется в количестве a_k, $k=1,2,\ldots,p$. Требуется выбрать наиболее экономичный комплекс средств для перевозки груза и объёмы груза, перевозимые каждой ёмкостью.

Математическая постановка задачи

Обозначим через x_j количество судов j-го типа, $j=1,2,\ldots,n$, выделяемых для перевозки, а через z_{ik} – количество груза вида k, подлежащего загрузке в емкость i, $k=1,2,\ldots,p$. Тогда математическая модель транспортной задачи о выборе средств доставки груза имеет вид:

$$y=\sum_{j=1}^{n} c_j x_j \to \min_{x_j\in\Omega}, \tag{1.35}$$

$$\Omega: \quad \sum_{j=1}^{n} d_{ij} x_j - \sum_{k=1}^{p} z_{ik} \ge 0, \quad i=\overline{1,m}, \tag{1.36}$$

$$\sum_{i=1}^{m} z_{ik} = a_k, \quad k=\overline{1,p}, \tag{1.37}$$

$$0 \le x_j \le q_j; \quad j=\overline{1,n}, \tag{1.38}$$

$$x_j = \mathrm{int}, \quad j=\overline{1,n}, \tag{1.39}$$

$$z_{ik} \ge 0; \quad i=\overline{1,m}; \quad k=\overline{1,p}. \tag{1.40}$$

Здесь целевая функция (1.35) определяет суммарные затраты на перевозку груза. Система ограничений в виде неравенств (1.36) показывает, что общее количество груза, загружаемое в ёмкости каждого типа i, не должно превышать суммарной грузоподъёмности этих ёмкостей по всем судам. Система двусторонней ограниченности переменных x_j (1.38) совместно с системами (1.39) и (1.40) оп-

ределяют пространство возможных решений в соответствии с условием задачи и физической природой переменных.

Пример транспортной задачи о выборе средств доставки груза

Пусть грузовой флот имеет в своем составе суда четырех типов. Количество судов *j*-го типа (*j=1,2,3,4*) соответственно равно 15, 20, 30, 25. Затраты при использовании одного судна *j*-го типа в планируемом периоде составляют соответственно 6, 5, 7, 4 ед. Каждое судно обладает грузовыми емкостями трех типов (трюмы, палубы, танки). Грузоподъемность d_{ij} каждой *i*–й емкости (*i=1,2,3*) на судне типа *j* определяется из матрицы $\mathbf{D} = \begin{bmatrix} 4 & 3 & 3 & 2 \\ 3 & 2 & 3 & 4 \\ 2 & 2 & 2 & 3 \end{bmatrix}$. Перевозке подлежит груз двух видов (*p*=2). Груз вида *k* (*k=1,2*) имеется соответственно в количестве 220 и 130. Требуется выбрать наиболее экономичный комплекс средств перевозки груза и объёмы груза, перевозимые каждой ёмкостью.

Математическая модель транспортной задачи о выборе средств доставки груза

Математическая модель задачи при использовании обозначений, принятых для общей модели задачи о выборе средств доставки груза (1.35) – (1.40), будет иметь вид:

$$y = 6x_1 + 5x_2 + 7x_3 + 4x_4 \to \min_{x_{ij} \in \Omega}$$

$$\Omega: \quad f_1 = 4x_1 + 3x_2 + 3x_3 + 2x_4 - (z_{11} + z_{12}) \geq 0,$$

$$f_2 = 3x_1 + 2x_2 + 3x_3 + 4x_4 - (z_{21} + z_{22}) \geq 0,$$

$$f_3 = 2x_1 + 2x_2 + 2x_3 + 3x_4 - (z_{31} + z_{32}) \geq 0,$$

$$f_4 = z_{12} + z_{22} + z_{32} = 220,$$

$$f_5 = z_{11} + z_{21} + z_{31} = 130,$$

$$0 \le x_1 \le 15,$$

$$0 \le x_2 \le 20,$$

$$0 \le x_3 \le 30,$$

$$0 \le x_4 \le 35,$$

$$x_j = \text{int}, \quad j = \overline{1,4},$$

$$z_{ik} \ge 0, \quad i = \overline{1,3}; \quad k = 1,2.$$

Цифровая модель и решение задачи в информационной среде *Microsoft Excel*

Распределение и назначение ячеек электронной таблицы для транспортной задачи о выборе средств доставки груза в условиях примера могут быть следующими:

- ячейки B6:E6 – для искомых переменных задачи x_j, $j = \overline{1,4}$;
- ячейки L5:M7 – для искомых переменных задачи z_{ik}, $i = \overline{1,3}$, $k = 1,2$;
- ячейки B4:E4 – для заданных количеств судов d_j каждого типа, $j = \overline{1,4}$;
- ячейки B9:E9 – для заданных затрат c_j на использование одного судна *j*–го типа, $j = \overline{1,4}$;
- ячейки B11:E11 – для промежуточных результатов с загруженными формулами для вычисления произведения $c_j x_j$, $j = \overline{1,4}$;
- ячейки G5:J7 – для указания грузоподъёмности транспортных средств d_{ij}, $i = \overline{1,3}$, $j = \overline{1,4}$;

- ячейки L9:M9 – для функций f_{3+k}, $k=\overline{1,2}$, с загруженными формулами для вычисления сумм $z_{1k}+z_{2k}+z_{3k}$;
- ячейки L11:M11 – для указания заданных объёмов вывозимого груза a_k, $k=1,2$;
- ячейки G14:J16 – для промежуточных результатов с загруженными формулами для вычисления произведения $d_{ij}x_j$, $i=\overline{1,3}$, $j=\overline{1,4}$;
- ячейки L14:L16 – для промежуточных результатов с загруженными формулами для вычисления сумм $d_{i1}x_1+d_{i2}x_2+d_{i3}x_3+d_{i4}x_4$, $i=\overline{1,3}$, ;
- ячейки N14:N16 – для промежуточных результатов с загруженными формулами для вычисления сумм $z_{1k}+z_{2k}$, $k=1,2$;
- ячейки P14:P16 – для окончательного вычисления функций f_i, $i=\overline{1,3}$, с соответствующими загруженными формулами;
- ячейка C13 – для значения целевой функции *y* с загруженной формулой для подсчёта суммы $\sum_{j=1}^{4} c_j x_j$.

Все ячейки, задействованные в цифровой модели, должны иметь *числовой* формат с числом десятичных знаков, равным 2. Исключение составляют ячейки с целочисленными данными. Это ячейки B4:E4 (для искомых переменных задачи) и B6:E6(для искомых переменных задачи)которые должны иметь числовой формат с числом десятичных знаков, равным 0.

После загрузки в электронную таблицу всех необходимых констант и формул для вычисления промежуточных и конечного результатов следует выполнить необходимые установки данных в диалоговом окне команды *Сервис/Поиск решения*. В условиях примера установки определяются следующим образом:

- для целевой ячейки – **$C$13**, равной **минимальному значению;**

- для ячеек с изменяемыми переменными – **$B$6:$E$6**;
  $L$5:$M$7;
- для ограничений: **$B$6:$E$6 >= 0**;
  $B$6:$E$6 = целое;
  $B$6:$E$6 <= $B$4: $E$4;
  $L$5:$M$7 >= 0;
  $P$14:$P$16 >= 0;
  $L$9:$M$9 = $L$11: $M$11.

На рис.1.18 показан вид экрана, который должен предшествовать выполнению команды *Сервис/Поиск решения*.

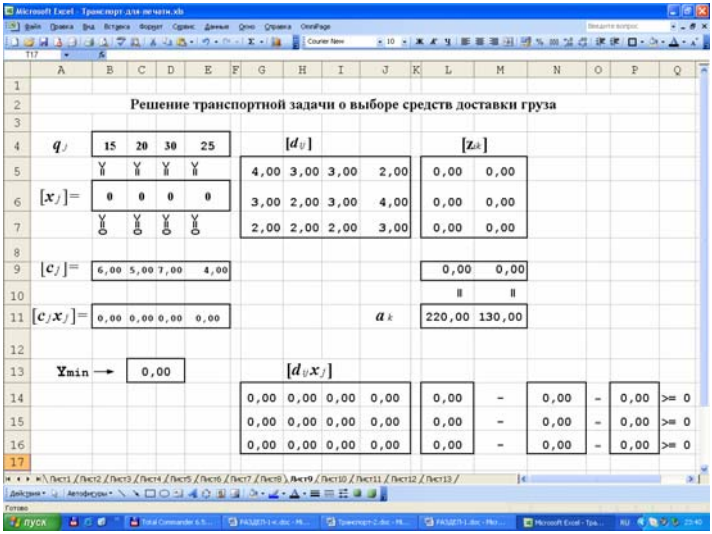
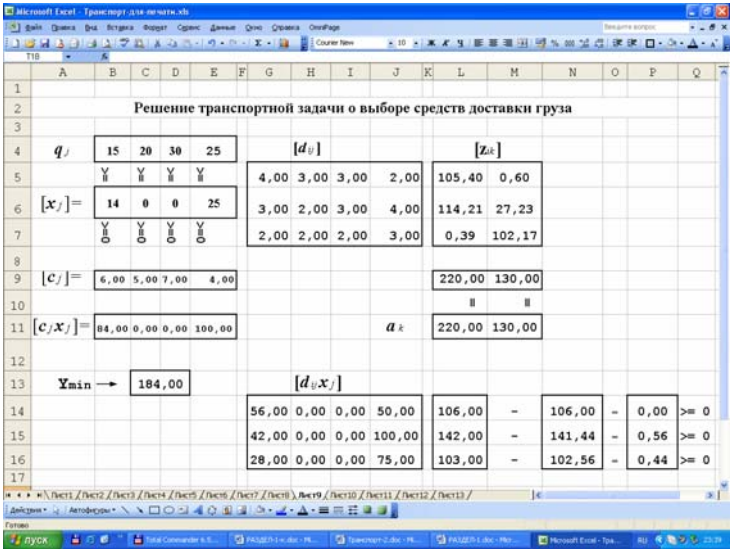


Рис. 1.18 – Экран с исходными установками

Вид экрана после выполнения команды *Сервис/Поиск решения* показан на рис. 1.19.



Решение транспортной задачи о выборе средств доставки груза

q_j		15	20	30	25		$[d_{ij}]$					$[z_{ik}]$					
		≥	≥	≥	≥		4,00	3,00	3,00	2,00		105,40	0,60				
$[x_j]=$		14	0	0	25		3,00	2,00	3,00	4,00		114,21	27,23				
		≥	≥	≥	≥		2,00	2,00	2,00	3,00		0,39	102,17				
$[c_j]=$		6,00	5,00	7,00	4,00							220,00	130,00				
												=	=				
$[c_j x_j]=$		84,00	0,00	0,00	100,00					a_k		220,00	130,00				
Xmin →		184,00					$[d_{ij} x_j]$										
							56,00	0,00	0,00	50,00		106,00	-	106,00	-	0,00	>= 0
							42,00	0,00	0,00	100,00		142,00	-	141,44	-	0,56	>= 0
							28,00	0,00	0,00	75,00		103,00	-	102,56	-	0,44	>= 0

Рис. 1.19 – Вид экрана с промежуточными и конечными результатами

Оптимальным решением задачи является вектор-строка

$$\mathbf{X}^{*T} = \begin{bmatrix} 14 & 0 & 0 & 25 \end{bmatrix}$$

и матрица

$$\mathbf{Z}^* = \begin{bmatrix} 105{,}4 & 0{,}6 \\ 114{,}21 & 27{,}23 \\ 0{,}39 & 102{,}17 \end{bmatrix},$$

которые обеспечивают минимальное значение целевой функции

$$y^* = 184 \text{ ус.ед.}$$

1.9. Транспортная задача о развозке груза

Данная задача, в отличие от всех других, рассматриваемых в данном разделе, относится к задачам организации перевозок по схеме «один ко многим». Она условно включена в группу транспортных задач, имеющих схему «многие ко многим».

Содержательная постановка задачи

Пусть некоторая центральная база снабжает продукцией (ее можно считать однородной) m складов. Развозка продукции на склады осуществляется одним грузовиком, причем каждый склад получает свой заказ полностью в один прием – грузоподъемность грузовика для этого достаточна. Грузовик может одновременно взять груз, соответствующий не более чем k заказам. Грузовик может объезжать склады по определенным r маршрутам. Один и тот же склад может находиться на разных маршрутах.

Пусть для каждого склада известна функция затрат в зависимости, например, от размера заказа. Требуется составить график развозок, обеспечивающий всех клиентов в полном объёме и минимизирующий суммарные затраты на перевозки. Время доставки никак не учитывается. Предполагается, что все операции по доставке заведомо могут быть осуществлены в течение некоторого периода времени, устраивающего всех потребителей.

Под способом развозки будем понимать любую допустимую комбинацию выполнения заказов. Он представляет собой m–мерный столбец, i-й компонент которого равен единице, если i–й заказ в этом способе удовлетворяется, и нулю – в противном случае. Для любой реальной задачи при небольших значениях m, k и r можно выписать все такие способы развозки. Число n этих способов будет зависеть не только от перечисленных параметров, но и от числа складов на каждом маршруте, объема заказов и т.д. Каждому j–му способу развозки соответствуют затраты c_j.

Пусть при данных конкретных условиях задачи составлена матрица $\mathbf{A}=[a_{ij}]$ всевозможных способов развозки, состоящее из нулей и единиц. Столбцы этой матрицы представляют собой описанные выше способы развозки, т.е. a_{ij} = 1, если в j–м способе i–й

заказ удовлетворяется, и a_{ij} = 0 в противном случае. Теперь задача состоит в выборе наиболее экономичной комбинации этих способов.

Математическая постановка задачи

Введем переменные

$$x_j = \begin{cases} 1, & \text{если } j\text{ - й способ перевозки реализуется,} \\ 0, & \text{в противном случае.} \end{cases} \tag{1.41}$$

Тогда математическая модель задачи принимает вид:

$$y = \sum_{j=1}^{n} c_j x_j \to \min_{x_j \in \Omega}, \tag{1.42}$$

$$\Omega: \quad \sum_{j=1}^{n} a_{ij} x_j = 1, \quad i = \overline{1,m}, \tag{1.43}$$

$$x_j \in \{0,1\}, \quad j = \overline{1,n}, \tag{1.44}$$

$$a_{ij} \in \{0,1\}, \quad i = \overline{1,m}, \quad j = \overline{1,j}. \tag{1.45}$$

Условие (1.43) означает, что все заказы должны быть удовлетворены только один раз. Выражения (1.44) и (1.45) определяют двоичный характер переменных x_j, $j = \overline{1,n}$, и элементов матрицы **A**.

Пример транспортной задачи о развозке груза

Пусть в рамках условий транспортной задачи о развозке груза известна матрица

$$\mathbf{A} = \left[a_{ij}\right] = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

всех возможных способов развозки груза из центральной базы в пять магазинов, а также затраты, связанные с реализацией каждого способа, а именно 11, 14, 9, 12, 13, 7,10, 8, 13 стоимостных ед. Составить график развозки, обеспечивающий минимальные суммарные затраты.

Математическая модель транспортной задачи о развозке груза в условиях примера

Математическая модель задачи при использовании обозначений, принятых для общей модели задачи о развозке груза (1.41) – (1.45), будет иметь вид:

$$y = 11x_1 + 14x_2 + 9x_3 + 12x_4 + 13x_5 +$$

$$+ 7x_6 + 10x_7 + 8x_8 + 13x_9 \to \min_{x_j \in \Omega},$$

$$\Omega: \quad f_1 = x_1 + x_3 + x_5 + x_6 + x_7 = 1,$$

$$f_2 = x_2 + x_5 + x_8 + x_9 = 1,$$

$$f_3 = x_2 + x_3 + x_4 + x_6 + x_8 = 1,$$

$$f_4 = x_1 + x_6 + x_8 + x_9 = 1,$$

$$f_5 = x_2 + x_3 + x_4 + x_5 + x_7 = 1,$$

$$x_j \in \{0,1\}, \quad j = \overline{1,9},$$

Цифровая модель и решение задачи в информационной среде *Microsoft Excel*

Распределение и назначение ячеек электронной таблицы для транспортной задачи о развозке груза в условиях примера могут быть следующими:

- ячейки B4:J4 – для искомых переменных задачи x_j, $j = \overline{1,9}$;
- ячейки B6:J10 – для элементов a_{ij}, $i = \overline{1,5}$, $j = \overline{1,9}$, матрицы **A;**
- ячейки B12:J12 – для указания затрат c_j на реализацию *j*–го способа, $j = \overline{1,9}$;
- ячейки B14:J14 – для промежуточных результатов с загруженными формулами для вычисления произведения $c_j x_j$, $j = \overline{1,9}$;
- ячейки B16:J20 – для промежуточных результатов с загруженными формулами для вычисления произведения $a_{ij} x_j$, $i = \overline{1,5}$, $j = \overline{1,9}$;
- ячейки L16:L20 – для вычисления функций f_i, $i = \overline{1,5}$, с соответствующими загруженными формулами;
- ячейка L8 – для значения целевой функции *y* с загруженной формулой для подсчёта суммы $\sum_{j=1}^{9} c_j x_j$.

Все ячейки, предназначенные для целочисленных данных, должны иметь *числовой* формат с числом десятичных знаков, равным 0. Это массивы ячеек B4:J4, B6:J10, B16:J20 и L16:L20. Остальные ячейки (L8, B12:J12 и B14:J14) должны иметь числовой формат с числом десятичных знаков, равным 2.

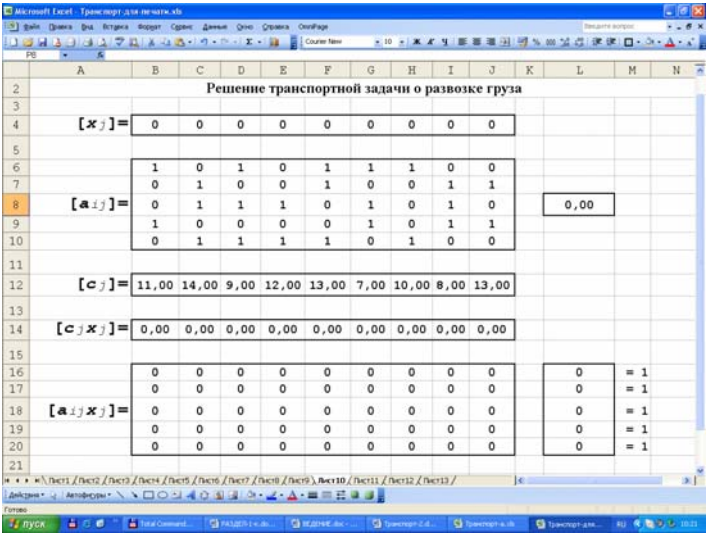
После загрузки в электронную таблицу всех необходимых констант и формул для вычисления промежуточных и конечного результатов следует выполнить необходимые установки данных в

диалоговом окне команды *Сервис/Поиск решения*. В условиях примера установки определяются следующим образом:

- для целевой ячейки – **$L$8**, равной **минимальному значению;**
- для ячеек с изменяемыми переменными – **$B$4:$J$4**;
- для ограничений: **$B$4:$J$4 = двоичное**;

  $L$16:$L$20 = 1.

На рис.1.20 показан вид экрана, который должен предшествовать выполнению команды *Сервис/Поиск решения*.

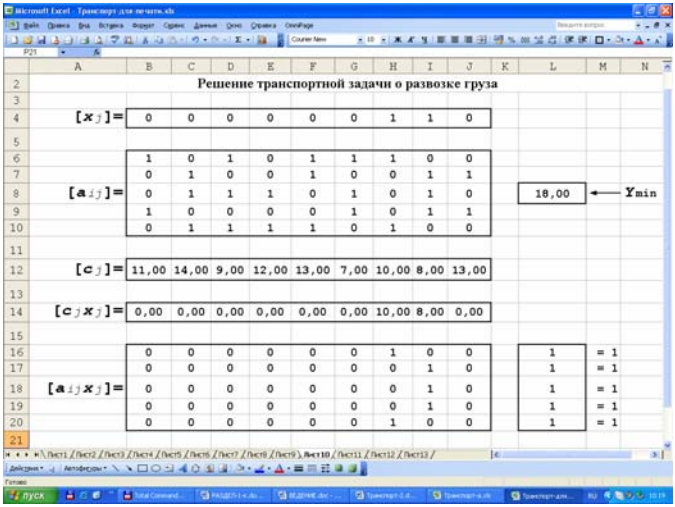


Решение транспортной задачи о развозке груза

[xj]=	0	0	0	0	0	0	0	0	0		
	1	0	1	0	1	1	1	0	0		
	0	1	0	0	1	0	0	1	1		
[aij]=	0	1	1	1	0	1	0	1	0	0,00	
	1	0	0	0	0	1	0	1	1		
	0	1	1	1	1	0	1	0	0		
[cj]=	11,00	14,00	9,00	12,00	13,00	7,00	10,00	8,00	13,00		
[cjxj]=	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00		
	0	0	0	0	0	0	0	0	0	0	= 1
	0	0	0	0	0	0	0	0	0	0	= 1
[aijxj]=	0	0	0	0	0	0	0	0	0	0	= 1
	0	0	0	0	0	0	0	0	0	0	= 1
	0	0	0	0	0	0	0	0	0	0	= 1

Рис. 1.20 – Экран с исходными установками

Вид экрана после выполнения команды *Сервис/Поиск решения* показан на рис. 1.21.



Решение транспортной задачи о развозке груза

[xj]=	0	0	0	0	0	0	1	1	0		
	1	0	1	0	1	1	1	0	0		
	0	1	0	0	1	0	0	1	1		
[aij]=	0	1	1	1	0	1	0	1	0	18,00	← Ymin
	1	0	0	0	0	1	0	1	1		
	0	1	1	1	1	0	1	0	0		
[cj]=	11,00	14,00	9,00	12,00	13,00	7,00	10,00	8,00	13,00		
[cjxj]=	0,00	0,00	0,00	0,00	0,00	0,00	10,00	8,00	0,00		
	0	0	0	0	0	0	1	0	0	1	= 1
	0	0	0	0	0	0	0	1	0	1	= 1
[aijxj]=	0	0	0	0	0	0	0	1	0	1	= 1
	0	0	0	0	0	0	0	1	0	1	= 1
	0	0	0	0	0	0	1	0	0	1	= 1

Рис. 1.21 – Вид экрана с промежуточными и конечными результатами

Оптимальным решением задачи является вектор-строка

$$X^{*T} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

которые обеспечивают минимальное значение целевой функции

$$y^* = 18 \ \textit{ед.}$$

1.10. Транспортная задача о двухэтапной перевозке груза

Содержательная постановка задачи

Однородный груз требуется доставить из m пунктов отправления в n пунктов назначения. При доставке в пункты назначения грузы могут быть сначала доставлены на p перевалочных пунктов. Заданы стоимости перевозок из каждого пункта отправления в каж-

дый пункт назначения c_{ij} ($i=\overline{1,m}, \quad j=\overline{1,n}$) и перевалочный пункт d_{ik} ($i=\overline{1,m}, \quad k=\overline{1,p}$), а также стоимости перевозки из каждого перевалочного пункта в каждый пункт назначения h_{kj} ($k=\overline{1,p}, \quad j=\overline{1,n}$). Требуется составить наиболее экономичный план перевозки грузов, обеспечивающий грузом пункты назначения в заданных объёмах. При этом товар по завершению перевозок должен быть полностью вывезен с перевалочных пунктов.

Математическая постановка задачи

Обозначим:

- a_i – запасы груза в i-м пункте отправления;
- b_j – потребность в грузе в j-м пункте назначения;
- c_k – вместимость k-го перевалочного пункта;
- c_{ij} – стоимость перевозки единицы груза из i-го пункта отправления в j-й пункт назначения, $i=\overline{1,m}, \quad j=\overline{1,n}$;
- d_{ik} – стоимость перевозки единицы груза из i-го пункта отправления в k-й перевалочный пункт $i=\overline{1,m}, \quad k=\overline{1,p}$;
- h_{kj} – стоимость перевозки единицы груза из k-го перевалочного пункта в j-й пункт назначения $k=\overline{1,p}, \quad j=\overline{1,n}$;
- x_{ij} – количество груза, перевозимого из i-го пункта отправления в j-й пункт назначения;
- y_{ik} – количество груза, перевозимого из i-го пункта отправления в k-й перевалочный пункт;
- z_{kj} – количество груза, перевозимого из k-го перевалочного пункта в j-й пункт назначения.

Математическая модель задачи с учетом выше приведенных обозначений может быть представлена в виде задачи линейного программирования:

$$s = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} \cdot x_{ij} + \sum_{i=1}^{m}\sum_{k=1}^{p} d_{ik} \cdot y_{ik} + \sum_{k=1}^{p}\sum_{j=1}^{n} h_{kj} \cdot z_{kj} \to \min_{x_{ij}, y_{ik}, z_{kj} \in \Omega}, \tag{1.46}$$

$$\Omega: \quad \sum_{j=1}^{n} x_{ij} + \sum_{k=1}^{p} y_{ik} \le a_i, \qquad i = \overline{1,m}, \tag{1.47}$$

$$\sum_{i=1}^{m} x_{ij} + \sum_{k=1}^{p} z_{kj} \ge b_j, \quad j = \overline{1,n}, \tag{1.48}$$

$$\sum_{i=1}^{m} y_{ik} \le c_k, \qquad k = \overline{1,p}, \tag{1.49}$$

$$\sum_{i=1}^{m} y_{ik} = \sum_{j=1}^{n} z_{kj}, \quad k = \overline{1,p} \tag{1.50}$$

$$x_{ijl} \ge 0; \; y_{ikl} \ge 0; \; z_{kjl} \ge 0; \; i = \overline{1,m}, \; j = \overline{1,n}, \; k = \overline{1,p}. \tag{1.51}$$

Здесь условие (1.47) ограничивает объёмы перевозимого груза возможностями заводов-изготовителей. Ограничение (1.48) обязывает компанию удовлетворить потребности всех баз в товаре. Ограничение (1.49) исключает нарушение условия вместимости складов. Выражения (1.50) требует полного освобождения складов после завершения перевозок. Неравенства (1.51) определяют область возможных значений переменных.

Пример транспортной задачи о двухэтапной перевозке груза

Компания располагает 2 заводами, 4 складами. Склады имеют вместимость 45000, 20000, 30000, 15000 *ед*. продукции, соответственно. Предприятие обеспечивает товарами пять торговых баз. Запросы потребителей составляют 30000, 23000, 15000, 32000, 16000 *ед*. продукции, соответственно. Задача предприятия – минимизация затрат на перевозку товаров от заводов к складам, от заводов к базам и от складов к базам. Затраты на перевозку единицы продукции от изготовителей на базы и склады, а также со складов на базы сведены в табл.3.1 Количество товара, полученного складом с заводов должно совпадать с количеством товара, который вывозится со склада на базы. Как спланировать перевозку товаров, если заводы могут произвести не более 60 000 ед. продукции каждый?

Таблица 3.1 – Тарифы перевозок единицы продукции

Отправители	Получатели				
	База 1	База 2	База 3	База 4	База 5
Завод 1	1.75	2.50	1.50	2.00	1.50
Завод 2	2.00	2.50	2.50	1.50	1.00
	Склад 1	Склад 2	Склад 3	Склад 4	
Завод 1	0.50	0.50	1.00	0.20	
Завод 2	1.50	0.30	0.50	0.20	
	База 1	База 2	База 3	База 4	База 5
Склад 1	1.50	1.50	0.50	1.50	3.00
Склад 2	1.00	0.50	0.50	1.00	0.50
Склад 3	1.00	1.50	2.00	2.00	0.50
Склад 4	2.50	1.50	0.20	1.50	0.50

Математическая модель транспортной задачи о двухэтапной перевозке груза в условиях примера

Математическая модель задачи при использовании обозначений, принятых для общей модели задачи о развозке груза (1.46) – (1.51), будет иметь вид:

$$
\begin{aligned}
s = {} & 1{,}75x_{11} + 2{,}5x_{12} + 1{,}5x_{13} + 2x_{14} + 1{,}5x_{15} + \\
& + 2x_{21} + 2{,}5x_{22} + 2{,}5x_{23} + 1{,}5x_{24} + x_{25} + \\
& + 0{,}5y_{11} + 0{,}5y_{12} + y_{13} + 0{,}2y_{14} + \\
& + 1{,}5y_{21} + 0{,}3y_{22} + 0{,}5y_{23} + 0{,}2y_{24} + \\
& + 1{,}5z_{11} + 1{,}5z_{12} + 0{,}5z_{13} + 1{,}5z_{14} + 3z_{15} + \\
& + z_{21} + 0{,}5z_{22} + 0{,}5z_{23} + z_{24} + 0{,}5z_{15} + \\
& + z_{31} + 1{,}5z_{32} + 2z_{33} + 2z_{34} + 0{,}5z_{35} + \\
& + 2{,}5z_{41} + 1{,}5z_{42} + 0{,}2z_{43} + 1{,}5z_{44} + 0{,}5z_{45} \to \min_{x_j \in \mathbf{\Omega}},
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{\Omega} : f_1 &= x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + y_{11} + y_{12} + y_{13} + y_{14} \le 60000, \\
f_2 &= x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + y_{21} + y_{22} + y_{23} + y_{24} \le 60000, \\
f_3 &= x_{11} + x_{21} + z_{11} + z_{21} + z_{31} + z_{41} \ge 30000, \\
f_4 &= x_{12} + x_{22} + z_{12} + z_{22} + z_{33} + z_{42} \ge 23000, \\
f_5 &= x_{13} + x_{23} + z_{13} + z_{23} + z_{33} + z_{43} \ge 15000, \\
f_6 &= x_{14} + x_{24} + z_{14} + z_{24} + z_{34} + z_{44} \ge 32000, \\
f_7 &= x_{15} + x_{25} + z_{15} + z_{25} + z_{35} + z_{45} \ge 16000, \\
f_8 &= y_{11} + y_{21} \le 45000, \\
f_9 &= y_{12} + y_{22} \le 20000, \\
f_{10} &= y_{13} + y_{23} \le 45000, \\
f_{11} &= y_{14} + y_{24} \le 15000,
\end{aligned}
$$

$$f_{12} = y_{11} + y_{21} = z_{11} + z_{12} + z_{13} + z_{14} + z_{15},$$
$$f_{13} = y_{12} + y_{22} = z_{21} + z_{22} + z_{23} + z_{24} + z_{25},$$
$$f_{14} = y_{13} + y_{23} = z_{31} + z_{32} + z_{33} + z_{34} + z_{35},$$
$$f_{15} = y_{14} + y_{24} = z_{41} + z_{42} + z_{43} + z_{44} + z_{45},$$
$$x_{ij} \geq 0;\ y_{ik} \geq 0;\ z_{kj} \geq 0,\ \ i = \overline{1,2},\ \ j = \overline{1,5},\ \ k = \overline{1,4}.$$

Цифровая модель и решение задачи в информационной среде *Microsoft Excel*

Распределение и назначение ячеек электронной таблицы для транспортной задачи о развозке груза в условиях примера могут быть следующими:

- ячейки B4:E5 – для заданных тарифов перевозки, d_{ik}, $i = 1,2,\quad k = \overline{1,4}$;
- ячейки B10:F11 – для заданных тарифов перевозки c_{ij}, $i = 1,2,\quad j = \overline{1,5}$;
- ячейки B16:F19 – для заданных тарифов перевозки h_{kj}, $k = \overline{1,4},\quad j = \overline{1,5}$;
- ячейки H4:K5 – для искомых переменных задачи y_{ik}, $i = 1,2,\quad k = \overline{1,4}$
- ячейки H10:L11 – для искомых переменных задачи x_{ij}, $i = 1,2,\quad j = \overline{1,5}$;
- ячейки N16:L19 – для искомых переменных задачи z_{kj}, $k = \overline{1,4},\quad j = \overline{1,5}$;

- ячейки N12:N13 – для указания возможностей заводов a_i, $i = 1,2$;
- ячейки H7:K7 – для указания вместимости складов c_k, $k = \overline{1,4}$;
- ячейки H21:L21 – для указания запросов от баз b_j, $j = \overline{1,5}$;
- ячейки L4:L5 – для промежуточных результатов с загрузкой формул для вычисления сумм $\sum_{k=1}^{4} y_{ik}$, $i = 1,2$;
- ячейки H6:K6 – для промежуточных результатов с загрузкой формул для вычисления сумм $\sum_{i=1}^{2} y_{ik}$, $k = \overline{1,4}$;
- ячейки M10:M11 – для промежуточных результатов с загрузкой формул для вычисления сумм $\sum_{i=1}^{2} x_{ij}$, $j = \overline{1,5}$;
- ячейки M12:M13 – для вычисления функций f_i, $i = 1,2$;
- ячейки M16:M19 – для промежуточных результатов с загрузкой формул для вычисления сумм $\sum_{j=1}^{5} z_{kj}$, $k = \overline{1,4}$;
- ячейки H20:L20 – для вычисления функций f_{2+j}, $j = \overline{1,5}$;
- ячейка L23 – для целевой функции *s* с соответствующей формулой для её вычисления.

Все ячейки электронной таблицы, задействованные в решении задачи должны иметь *числовой* формат с числом десятичных знаков, равным 2.

После загрузки в электронную таблицу всех необходимых констант и формул для вычисления промежуточных и конечного ре-

зультатов следует выполнить необходимые установки данных в диалоговом окне команды *Сервис/Поиск решения*. В условиях примера установки определяются следующим образом:

- для целевой ячейки – **$I$23**, равной **минимальному значению;**
- для ячеек с изменяемыми переменными – **$H$4:$K$5**; **$H$10:$L$11; $H$16:$L$19;**
- для ограничений: **$H$10:$L$11 >= 0**;
  $H$16:$L$19 >= 0;
  $H$4:$K$5 >= 0;
  $H$20:$L$20 >= $H$21:$L$21;
  $H$6:$K$6 < = $H$7:$K$7;
  $H$6:$K$6 = $M$16:$M$19;
  $M$12:$M$13 <= $N$12:$N$13.

На рис.1.22 показан вид экрана, который должен предшествовать выполнению команды *Сервис/Поиск решения*.

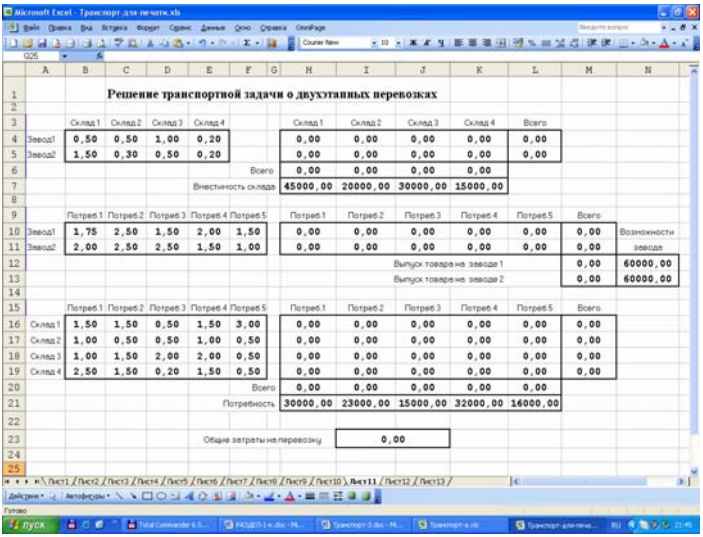
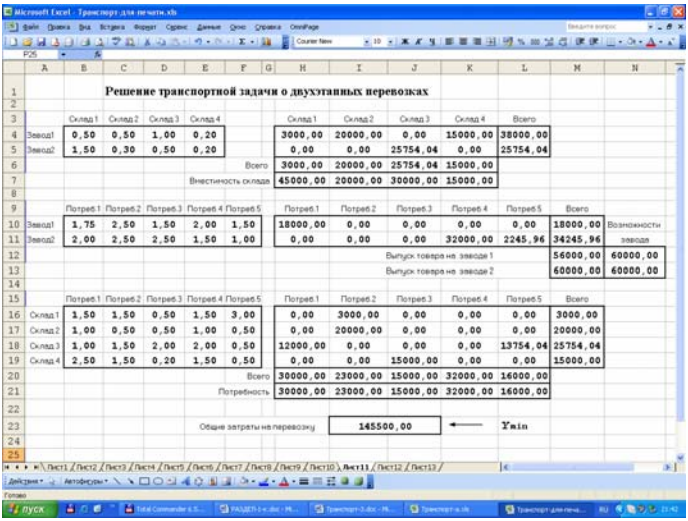


Рис. 1.22 – Экран с исходными установками

Вид экрана после выполнения команды *Сервис/Поиск решения* показан на рис. 1.23.



Решение транспортной задачи о двухэтапных перевозках

	Склад 1	Склад 2	Склад 3	Склад 4
Завод1	0,50	0,50	1,00	0,20
Завод2	1,50	0,30	0,50	0,20

	Склад 1	Склад 2	Склад 3	Склад 4	Всего
Завод1	3000,00	20000,00	0,00	15000,00	38000,00
Завод2	0,00	0,00	25754,04	0,00	25754,04
Всего	3000,00	20000,00	25754,04	15000,00	
Внеместность склада	45000,00	20000,00	30000,00	15000,00	

	Потреб.1	Потреб.2	Потреб.3	Потреб.4	Потреб.5
Завод1	1,75	2,50	1,50	2,00	1,50
Завод2	2,00	2,50	2,50	1,50	1,00

	Потреб.1	Потреб.2	Потреб.3	Потреб.4	Потреб.5	Всего	
Завод1	18000,00	0,00	0,00	0,00	0,00	18000,00	Возможности завода
Завод2	0,00	0,00	0,00	32000,00	2245,96	34245,96	
Выпуск товара на заводе 1						56000,00	60000,00
Выпуск товара на заводе 2						60000,00	60000,00

	Потреб.1	Потреб.2	Потреб.3	Потреб.4	Потреб.5
Склад 1	1,50	1,50	0,50	1,50	3,00
Склад 2	1,00	0,50	0,50	1,00	0,50
Склад 3	1,00	1,50	2,00	2,00	0,50
Склад 4	2,50	1,50	0,20	1,50	0,50

	Потреб.1	Потреб.2	Потреб.3	Потреб.4	Потреб.5	Всего
Склад 1	0,00	3000,00	0,00	0,00	0,00	3000,00
Склад 2	0,00	20000,00	0,00	0,00	0,00	20000,00
Склад 3	12000,00	0,00	0,00	0,00	13754,04	25754,04
Склад 4	0,00	0,00	15000,00	0,00	0,00	15000,00
Всего	30000,00	23000,00	15000,00	32000,00	16000,00	
Потребность	30000,00	23000,00	15000,00	32000,00	16000,00	

Общие затраты на перевозку: 145500,00 ← Ymin

Рис. 1.23 – Вид экрана с промежуточными и конечными результатами

Оптимальным решением задачи являются матрицы

$$\mathbf{X}^* = \begin{bmatrix} 18000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 32000 & 2246 \end{bmatrix},$$

$$\mathbf{Y}^* = \begin{bmatrix} 3000 & 20000 & 0 & 15000 \\ 0 & 0 & 25754 & 0 \end{bmatrix},$$

$$\mathbf{Z}^* = \begin{bmatrix} 0 & 3000 & 0 & 0 & 0 \\ 0 & 20000 & 0 & 0 & 0 \\ 12000 & 0 & 0 & 0 & 13754 \\ 0 & 0 & 15000 & 0 & 0 \end{bmatrix},$$

которые обеспечивают минимальное значение целевой функции

$$y^* = 145500 \text{ ед.}$$

1.11. Транспортная задача о двухэтапной перевозке груза нескольких видов

Содержательная постановка задачи

Груз, включающий q видов продукции, требуется доставить из m пунктов отправления в n пунктов назначения. При доставке в пункты назначения грузы могут быть сначала доставлены на p перевалочных пунктов. Заданы стоимости перевозок для каждого вида груза из каждого пункта отправления в каждый пункт назначения и перевалочный пункт, а также стоимости перевозки из каждого перевалочного пункта в каждый пункт назначения.

Математическая модель задачи

Обозначим:

- a_{il} – запасы l-го вида груза в i-м пункте отправления;
- b_{jl} – потребность в l-м виде груза в j-м пункте назначения;
- c_{kl} – вместимость k-го перевалочного пункта по отношению к l-му виду груза;
- c_{ijl} – стоимость перевозки единицы l-го вида груза из i-го пункта отправления в j-й пункт назначения $i = \overline{1,m}$, $j = \overline{1,n}$, $l = \overline{1,q}$;
- c^*_{ikl} – стоимость перевозки единицы l-го вида груза из i-го пункта отправления в k-й перевалочный пункт $i = \overline{1,m}$, $k = \overline{1,p}$, $l = \overline{1,q}$;

- c^{**}_{kjl} – стоимость перевозки единицы l-го вида груза из k-го перевалочного пункта в j-й пункт назначения $k=\overline{1,p}$, $j=\overline{1,n}$, $l=\overline{1,q}$;
- x_{ijl} – количество l-го вида груза, перевозимого из i-го пункта отправления в j-й пункт назначения;
- y_{ikl} – количество l-го вида груза, перевозимого из i-го пункта отправления в k-й перевалочный пункт;
- z_{kjl} – количество l-го вида груза, перевозимого из k-го перевалочного пункта в j-й пункт назначения.

Математическая модель задачи с учетом выше приведенных обозначений может быть представлена в виде задачи линейного программирования:

$$y=\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{l=1}^{q}c_{ijl}\cdot x_{ijl}+\sum_{i=1}^{m}\sum_{k=1}^{p}\sum_{l=1}^{q}c^{*}_{ikl}\cdot y_{ikl}+$$

$$+\sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q}c^{**}_{kjl}\cdot z_{kjl}\to\min_{x_{ijl},y_{ikl},z_{kjl}\in\Omega}, \qquad (1.52)$$

$$\Omega:\quad \sum_{j=1}^{n}x_{ijl}+\sum_{k=1}^{p}y_{ikl}\le a_{il},\quad i=\overline{1,m},\quad l=\overline{1,q}, \qquad (1.53)$$

$$\sum_{i=1}^{m}x_{ijl}+\sum_{k=1}^{p}z_{kjl}\ge b_{jl},\quad j=\overline{1,n},\quad l=\overline{1,q}, \qquad (1.54)$$

$$\sum_{i=1}^{m}y_{ikl}\le c_{kl},\quad k=\overline{1,p},\quad l=\overline{1,q}, \qquad (1.55)$$

$$\sum_{i=1}^{m}y_{ikl}=\sum_{j=1}^{n}z_{kjl},\quad k=\overline{1,p},\quad l=\overline{1,q}, \qquad (1.56)$$

$$x_{ijl} \geq 0;\ \ y_{ikl} \geq 0;\ z_{kjl} \geq 0;\ i = \overline{1,m},\ j = \overline{1,n}\ ,\ k = \overline{1,p}\ ,\ l = \overline{1,q}\ . \quad (1.57).$$

Из-за громоздкости математических моделей конкретный пример транспортной задачи о двухэтапной перевозке груза нескольких видов не приводится.

1.12. Транспортная задача о двухэтапной перевозке груза нескольких видов по запросам потребителей

Содержательная постановка задачи

Существует модификация транспортной задачи двухэтапной перевозки грузов, в которой количество груза в пунктах отправления не фиксировано, а зависит от запросов потребителей.

Математическая модель задачи

Обозначим:

- a_{il} – количество производимого l-го вида груза в i-м пункте отправления;
- t_{il} – затраты на производство l-го вида груза в i-м пункте отправления;
- b_{jl} – потребность в l-м виде груза в j-м пункте назначения;
- c_{kl} – вместимость k-го перевалочного пункта по отношению к l-му виду груза ;
- c_{ijl} – стоимость перевозки единицы l-го вида груза из i-го пункта отправления в j-й пункт назначения $i = \overline{1,m},\ \ j = \overline{1,n},$ $l = \overline{1,q}$;
- c^{*}_{ikl} – стоимость перевозки единицы l-го вида груза из i-го пункта отправления в k-й перевалочный пункт $i = \overline{1,m},\ \ k = \overline{1,p},$ $l = \overline{1,q}$;

- c^{**}_{kjl} – стоимость перевозки единицы l-го вида груза из k-го перевалочного пункта в j-й пункт назначения $k=\overline{1,p},\ j=\overline{1,n},$ $l=\overline{1,q}$;
- b_{jl} – потребность в l-м виде груза в j-м пункте назначения;
- s_{il} – количество производимого l-го вида груза в i-м пункте отправления;
- x_{ijl} – количество l-го вида груза, перевозимого из i-го пункта отправления в j-й пункт назначения;
- y_{ikl} – количество l-го вида груза, перевозимого из i-го пункта отправления в k-й перевалочный пункт;
- z_{kjl} – количество l-го вида груза, перевозимого из k-го перевалочного пункта в j-й пункт назначения.

Математическая модель задачи с учетом выше приведенных обозначений может быть представлена в виде задачи линейного программирования:

$$y=\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{l=1}^{q}c_{ijl}\cdot x_{ijl}+\sum_{i=1}^{m}\sum_{k=1}^{p}\sum_{l=1}^{q}c^{*}_{ikl}\cdot y_{ikl}+$$

$$+\sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q}c^{**}_{kjl}\cdot z_{kjl}+\sum_{i=1}^{m}\sum_{l=1}^{q}t_{il}s_{il}\to\min_{x_{ijl},y_{ikl},z_{kjl},s_{il}\in\Omega},\quad (1.58)$$

$$\Omega:\quad \sum_{j=1}^{n}x_{ijl}+\sum_{k=1}^{p}y_{ikl}\le s_{il},\ \ i=\overline{1,m},\ \ l=\overline{1,q},\quad (1.59)$$

$$\sum_{i=1}^{m}x_{ijl}+\sum_{k=1}^{p}z_{kjl}\ge b_{jl},\ \ j=\overline{1,n},\ l=\overline{1,q},\quad (1.60)$$

$$\sum_{i=1}^{m}y_{ikl}\le c_{kl},\ k=\overline{1,p}\ ,\ \ l=\overline{1,q},\quad (1.61)$$

$$\sum_{i=1}^{m} y_{ik} = \sum_{j=1}^{n} z_{kj}\ ,\ k=\overline{1,p}\ ,\ l=\overline{1,q}, \qquad (1.62)$$

$$x_{ijl} \ge 0;\ y_{ikl} \ge 0;\ z_{kjl} \ge 0;\ s_{il} \ge 0;\ i=\overline{1,m},\ j=\overline{1,n}\ ,\ k=\overline{1,p}\ ,\ l=\overline{1,q}\ . \qquad (1.63)$$

Из-за громоздкости математических моделей конкретный пример транспортной задачи о двухэтапной перевозке груза нескольких видов по запросам пользователей не приводится.

1.13. Транспортная задача о закрытии завода

Содержательная постановка задачи

Производственное объединение располагает m заводами и n складами. Заданы потребности складов в продукте и стоимости на перевозку продуктов с каждого завода на каждый склад. Также заданы фиксированные стоимости функционирования заводов, заданы возможности заводов по производству продукта. Производственное объединение рассматривает возможность закрытия одного или нескольких заводов. Это должно уменьшить затраты на перевозку. Какие заводы, если это целесообразно, должны быть закрыты?

Математическая модель задачи

Обозначим:

- c_{ij} – стоимости перевозки с j-го завода на i-й склад, $i=\overline{1,n},\ \ j=\overline{1,m}$;
- d_i – потребности i-го склада в продукте, $i=\overline{1,n}$;
- a_j – возможность j-го завода по производству продукта, $j=\overline{1,m}$;

- e_j – фиксированная стоимость функционирования j-го завода, $j = \overline{1,m}$;
- z_j – двоичное число, показывающее, нужно ли закрыть j-й завод (значение 0) или оставить его работать (значение 1), $j = \overline{1,m}$;
- x_{ij} – количество перевозимого товара с j-го завода на i-й склад, $i = \overline{1,n}$, $j = \overline{1,m}$.

Тогда математическая модель транспортной задачи о закрытии завода может быть представлена в виде:

$$y = \sum_{j=1}^{n}\left(e_j z_j + \sum_{i=1}^{m} c_{ij} x_{ij}\right) \to \min_{z_i, x_{ij} \in \Omega}, \tag{1.64}$$

$$\Omega: \quad \sum_{i=1}^{n} x_{ij} \le z_j a_j, \quad j = \overline{1,m}, \tag{1.65}$$

$$\sum_{j=1}^{m} x_{ij} \ge d_i, \quad i = \overline{1,n}, \tag{1.66}$$

$$x_{ij} \ge 0, \quad i = \overline{1,n}; \quad j = \overline{1,m}, \tag{1.67}$$

$$z_j \in \{0,1\}; \quad j = \overline{1,m}. \tag{1.68}$$

Здесь целевая функция (1.64) определяет общие затраты производственного объединения на функционирование заводов и транспортировку готовой продукции на склады. Ограничение (1.65) определяет возможности заводов по производству продукции. Ограничение (1.66) определяют потребности складов в готовой продукции. Требования (1.67) и (1.68) обусловлены физической сущностью соответствующих величин.

Пример транспортной задачи о закрытии завода

Производственное объединение включает 5 заводов и 4 склада. Заданы потребности складов в продукте, соответственно 15, 18, 14, 20 *ед*, и матрица стоимости на перевозку продуктов с каждого завода на каждый склад (в *ус. ед*).

$$\left[c_{ij}\right]=\begin{bmatrix} 40 & 20 & 30 & 30 & 40 \\ 25 & 30 & 25 & 30 & 30 \\ 12 & 18 & 26 & 41 & 30 \\ 22 & 26 & 31 & 37 & 32 \end{bmatrix}.$$

Также заданы фиксированные стоимости функционирования заводов, соответственно 120, 150, 170, 160, 130 *ус. ед*, и их возможности по производству продукта, соответственно 20, 22, 17, 19, 18 *ед*. Производственное объединение рассматривает возможность закрытия одного или нескольких заводов. Это должно уменьшить затраты на перевозку. Какие заводы, если это целесообразно, должны быть закрыты?

Математическая модель транспортной задачи о закрытии предприятия в условиях примера

Математическая модель задачи при использовании обозначений, принятых для общей модели задачи о развозке груза (1.64) – (1.68), будет иметь вид:

$$\begin{aligned} y = {} & 120z_1 + 40x_{11} + 25x_{21} + 12x_{31} + 22x_{41} + \\ & + 150z_2 + 20x_{12} + 30x_{22} + 18x_{32} + 26x_{42} + \\ & + 170z_3 + 30x_{13} + 25x_{23} + 26x_{33} + 31x_{43} + \\ & + 160z_4 + 30x_{14} + 30x_{24} + 41x_{34} + 37x_{44} + \end{aligned}$$

$$+130z_5 + 40x_{15} + 30x_{25} + 30x_{35} + 32x_{45} \to \min_{z_i, x_{ij} \in \Omega},$$

$$\Omega: \quad f_1 = x_{11} + x_{21} + x_{31} + x_{41} \le 20z_1,$$

$$f_2 = x_{12} + x_{22} + x_{32} + x_{42} \le 22z_2,$$

$$f_3 = x_{13} + x_{23} + x_{33} + x_{43} \le 17z_3,$$

$$f_4 = x_{14} + x_{24} + x_{34} + x_{44} \le 19z_4,$$

$$f_5 = x_{15} + x_{25} + x_{35} + x_{45} \le 18z_5,$$

$$f_6 = x_{11} + x_{12} + x_{13} + x_{14} + x_{15} \ge 15,$$

$$f_7 = x_{21} + x_{22} + x_{23} + x_{24} + x_{25} \ge 18,$$

$$f_8 = x_{31} + x_{32} + x_{33} + x_{34} + x_{35} \ge 14,$$

$$f_9 = x_{41} + x_{42} + x_{43} + x_{44} + x_{45} \ge 20,$$

$$x_{ij} \ge 0, \quad i = \overline{1,4}; \quad j = \overline{1,5},$$

$$z_j \in \{0,1\}; \quad j = \overline{1,5}.$$

Цифровая модель и решение задачи в информационной среде *Microsoft Excel*

Распределение и назначение ячеек электронной таблицы для транспортной задачи о развозке груза в условиях примера могут быть следующими:

- ячейки B5:F8 – для искомых переменных задачи $x_{ij}, i = \overline{1,4}, \ j = \overline{1,5}$;
- ячейки B3:F3 – для искомых переменных задачи z_j, $j = \overline{1,5}$;
- ячейки B10:F13 – для заданных тарифов перевозки, c_{ij}, $i = \overline{1,4}, \ j = \overline{1,5}$;

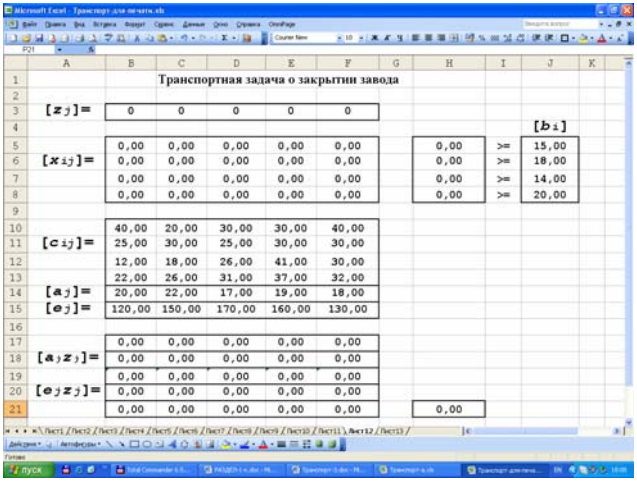
- ячейки B14:F14 – для указания возможностей завода a_j, $j=\overline{1,5}$;
- ячейки B15:F15 – для указания фиксированных стоимостей функционирования завода e_j, $j=\overline{1,5}$;
- ячейки B17:F17 – для вычисления функции f_j, $j=\overline{1,5}$, с загрузкой формулы для вычисления суммы $\sum_{i=1}^{4} x_{ij}$;
- ячейки H5:H8 – для вычисления функции f_{5+i}, $i=\overline{1,4}$, с загрузкой формулы для вычисления суммы $\sum_{j=1}^{5} x_{ij}$;
- ячейки B18:F18 – для промежуточных результатов с загрузкой формулы для вычисления произведения $z_j a_j$, $j=\overline{1,5}$;
- ячейки B19:F198 – для промежуточных результатов с загруженной формулой для вычисления суммы $\sum_{i=1}^{4} c_{ij} x_{ij}$, $j=\overline{1,5}$;
- ячейки B20:F20 – для промежуточных результатов с загрузкой формулы для вычисления произведения $e_j z_j$, $j=\overline{1,5}$;
- ячейки B21:F21 – для промежуточных результатов с загрузкой формулы для вычисления суммы $e_j z_j + \sum_{i=1}^{4} c_{ij} x_{ij}$, $j=\overline{1,5}$ произведения $z_j a_j$, $j=\overline{1,5}$;
- ячейки J5:J9 – для указания потребности склада в продукте d_i, $i=\overline{1,5}$
- ячейка H21 – для целевой функции *y* с соответствующей загруженной формулой для её вычисления.

Все ячейки электронной таблицы, задействованные в решении задачи должны иметь *числовой* формат с числом десятичных знаков, равным 2. Исключение составляют ячейки B3:F3, формат которых должен соответствовать двоичным числам.

После загрузки в электронную таблицу всех необходимых констант и формул для вычисления промежуточных и конечного результатов следует выполнить необходимые установки данных в диалоговом окне команды *Сервис/Поиск решения*. В условиях примера установки определяются следующим образом:

- для целевой ячейки – **$H$21**, равной **минимальному значению;**
- для ячеек с изменяемыми переменными – **$B$3:$F$3;**
  $B$5:$F$8;
- для ограничений: **$B$3:$F$3 = двоичное**;
  $B$5: $F$8 >= 0;
  $H$5:$H$9 >= $J$5:$J$9;
  $B$17:$F$17 <= $B$18:$F$18.

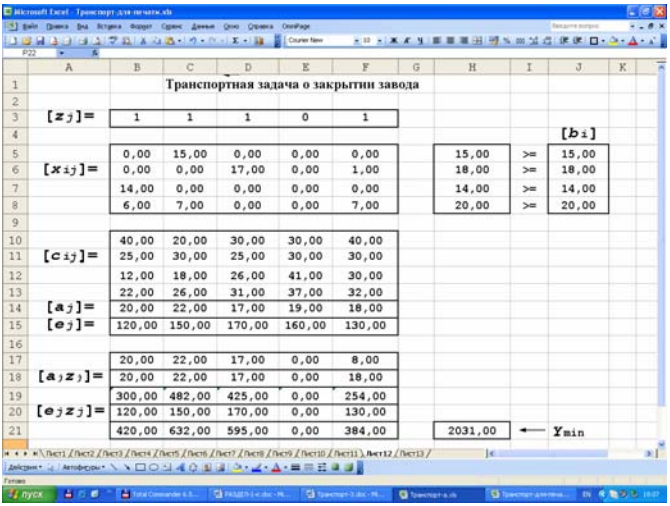
На рис.1.24 показан вид экрана, который должен предшествовать выполнению команды *Сервис/Поиск решения*.



	A	B	C	D	E	F	G	H	I	J	K
1		Транспортная задача о закрытии завода									
2											
3	[zj]=	0	0	0	0	0					
4										[bi]	
5		0,00	0,00	0,00	0,00	0,00		0,00	>=	15,00	
6	[xij]=	0,00	0,00	0,00	0,00	0,00		0,00	>=	18,00	
7		0,00	0,00	0,00	0,00	0,00		0,00	>=	14,00	
8		0,00	0,00	0,00	0,00	0,00		0,00	>=	20,00	
9											
10		40,00	20,00	30,00	30,00	40,00					
11	[cij]=	25,00	30,00	25,00	30,00	30,00					
12		12,00	18,00	26,00	41,00	30,00					
13		22,00	26,00	31,00	37,00	32,00					
14	[aj]=	20,00	22,00	17,00	19,00	18,00					
15	[ej]=	120,00	150,00	170,00	160,00	130,00					
16											
17		0,00	0,00	0,00	0,00	0,00					
18	[ajzj]=	0,00	0,00	0,00	0,00	0,00					
19		0,00	0,00	0,00	0,00	0,00					
20	[ejzj]=	0,00	0,00	0,00	0,00	0,00					
21		0,00	0,00	0,00	0,00	0,00		0,00			

Рис. 1.24 – Экран с исходными установками

Вид экрана после выполнения команды *Сервис/Поиск решения* показан на рис. 1.25.



Транспортная задача о закрытии завода

[zj]=	1	1	1	0	1				
									[bi]
	0,00	15,00	0,00	0,00	0,00	15,00	>=	15,00
[xij]=	0,00	0,00	17,00	0,00	1,00	18,00	>=	18,00
	14,00	0,00	0,00	0,00	0,00	14,00	>=	14,00
	6,00	7,00	0,00	0,00	7,00	20,00	>=	20,00
	40,00	20,00	30,00	30,00	40,00			
[cij]=	25,00	30,00	25,00	30,00	30,00			
	12,00	18,00	26,00	41,00	30,00			
	22,00	26,00	31,00	37,00	32,00			
[aj]=	20,00	22,00	17,00	19,00	18,00			
[ej]=	120,00	150,00	170,00	160,00	130,00			
	20,00	22,00	17,00	0,00	8,00			
[ajzj]=	20,00	22,00	17,00	0,00	18,00			
	300,00	482,00	425,00	0,00	254,00			
[ejzj]=	120,00	150,00	170,00	0,00	130,00			
	420,00	632,00	595,00	0,00	384,00	2031,00	←	Ymin

Рис. 1.25 – Вид экрана с промежуточными и конечными результатами

Оптимальным решением задачи являются вектор-строка

$$\mathbf{Z}^{*\mathrm{T}} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

и матрица

$$\mathbf{Z}^{*} = \begin{bmatrix} 0 & 15 & 0 & 0 & 0 \\ 0 & 0 & 17 & 0 & 1 \\ 14 & 0 & 0 & 0 & 0 \\ 6 & 7 & 0 & 0 & 7 \end{bmatrix},$$

которые обеспечивают минимальное значение целевой функции.

Анализ решения задачи позволяет сделать вывод, что закрытие 4-го завода приведет к сокращению затрат на перевозку продукции без нарушения заданных поставок продукта.

1.14. Транспортная задача о розыгрыше кубка

Содержательная постановка задачи

Спортивный кубок разыгрывают 2^n клубных команд ($n = 1,2,\ldots$), тренировочные базы которых находятся в разных городах. Соревнования проводятся по системе *play-off*. В каждом туре встречаются две команды, одна из которых по результатом двух игр («в гостях» и «дома») выбывает из дальнейшей борьбы. Известны расстояния между городами. Как организаторы соревнований должны составить пары команд на первый тур, чтобы суммарное расстояние при переездах команд из города в город было минимальным?

Математическая модель задачи

Обозначим:

- l_{ij} – расстояние между i-м и j-м городами, $i = \overline{1,2^n}$, $j = \overline{1,2^n}$;
- x_{ij} – переменная, принимающая значение 1, если i-я команда в туре играет с j-й командой, и значение 0 – в противном случае, $i = \overline{1,2^n}$, $j = \overline{1,2^n}$.

Математическая модель задачи имеет вид:

$$y = 2\sum_{i=1}^{2^n-1}\sum_{j=i+1}^{2^n} l_{ij}x_{ij} \to \min_{x_{ij}\in\Omega}, \tag{1,69}$$

$$\Omega: \quad \sum_{\substack{i=1,\\ i\neq j}}^{2^n} x_{ij} = 1; \quad j = \overline{1,2^n}, \tag{1.70}$$

$$\sum_{\substack{j=1,\\ j\neq i}}^{2^n} x_{ij} = 1, \quad i = \overline{1,2^n}, \tag{1.71}$$

$$x_{ij} \in \{0, 1\}, \ i = \overline{1,2^n}, \ j = \overline{1,2^n}. \qquad (1.72)$$

$$x_{ij} = x_{ji}, \ i = \overline{1,2^n}, \ j = \overline{1,2^n}. \qquad (1.73)$$

Здесь целевая функция (1.69) определяет общие затраты на проведение соревнований (при моделировании затрат учитывалось, что расстояния из пункта А в В равно расстоянию из пункта В в А, а также, тот факт, что команда не может играть сама с собой. Ограничение (1.70) говорит о том, что *j*-я команда должна сыграть с одной из ($2^n - 1$) оставшихся команд. Аналогично ограничение (1.71) говорит о том, что *i*-я команда должна сыграть с одной из ($2^n - 1$) оставшихся команд. Выражение (1.72) определяет возможные значения переменных. Равенство (1.73) фиксирует тот факт, что если *j*-я команда встречается с *i*-й, то *i*-я команда, само собой, встречается с *j*-й.

Пример транспортной задачи о розыгрыше кубка

В очередном туре розыгрыша спортивного кубка принимают участие 8 команд. Соревнования проводятся по системе *play-off*. Команда, которая выбывает из соревнований, определяется по результатам двух игр: «в гостях» и «дома». Задана матрица расстояний в *км* между городами-участниками тура:

$$\mathbf{L} = \begin{Bmatrix} 0 & 150 & 400 & 370 & 120 & 600 & 350 & 200 \\ 155 & 0 & 180 & 500 & 420 & 280 & 220 & 530 \\ 400 & 180 & 0 & 370 & 450 & 280 & 310 & 440 \\ 370 & 500 & 370 & 0 & 300 & 220 & 360 & 530 \\ 120 & 420 & 450 & 300 & 0 & 140 & 270 & 540 \\ 600 & 280 & 280 & 220 & 140 & 0 & 390 & 170 \\ 350 & 220 & 310 & 360 & 270 & 390 & 0 & 270 \\ 200 & 530 & 440 & 530 & 540 & 170 & 270 & 0 \end{Bmatrix}.$$

Разбить команды по парам для участия в играх очередного тура так, чтобы суммарное расстояние переезда команд для проведения всех игр тура было минимальным.

Математическая модель транспортной задачи о розыгрыше кубка в условиях примера

Математическая модель задачи при использовании обозначений, принятых для общей модели задачи о розыгрыше спортивного кубка (1.69) – (1.73), будет иметь вид:

$$y = 2\big(150x_{12} + 400x_{13} + 370x_{14} + 120x_{15} + 600x_{16} + 350x_{17} +$$
$$+ 200x_{18} + 180x_{23} + 500x_{24} + 420x_{25} + 280x_{26} + 220x_{27} +$$
$$+ 530x_{28} + 370x_{34} + 450x_{35} + 280x_{36} + 310x_{37} + 440x_{38} +$$
$$+ 300x_{45} + 220x_{46} + 360x_{47} + 530x_{48} + 140x_{56} + 270x_{57} +$$
$$+ 540x_{58} + 390x_{67} + 170x_{68} + 270x_{78}\big) \to \min_{x_{ij} \in \Omega},$$

$$\Omega: \quad \sum_{\substack{i=1,\\ i \neq j}}^{8} x_{ij} = 1, \quad j = \overline{1,8},$$

$$\sum_{\substack{j=1,\\ j \neq i}}^{8} x_{ij} = 1, \quad i = \overline{1,8},$$

$$x_{ij} \in \{0, 1\}, \quad i = \overline{1,8}, \quad j = \overline{1,8}, \quad i \neq j,$$

$$x_{ij} = x_{ji}, \quad i = \overline{1,8}, \quad j = \overline{1,8}, \quad i \neq j.$$

Цифровая модель и решение задачи в информационной среде *Microsoft Excel*

Распределение и назначение ячеек электронной таблицы для транспортной задачи о розыгрыше спортивного кубка в условиях примера могут быть следующими:

- ячейки D4:J4, C5, E5:J5, C6:D6, F6:J6, C7:E7, G7:J7, C8:F8, H8:J8, C9:G9, I9:J9, C10:H10, J10, C11:I11 – для искомых переменных задачи $x_{ij}, i = \overline{1,8}, \quad j = \overline{1,8}, \quad i \neq j$;

- ячейки C13:J13 – для вычисления суммы $\sum_{\substack{i=1,\\ i\neq j}}^{8} x_{ij} = 1,$ $j = \overline{1,8}$, с загрузкой соответствующей формулы;
- ячейки L4:L11 – для вычисления суммы $\sum_{\substack{j=1,\\ j\neq i}}^{8} x_{ij} = 1,\ i = \overline{1,8},$ с загрузкой соответствующей формулы;
- ячейки D15:J15, E16:J16, F17:J17, G18:J18, H19:J19, I20:J20, J21 – для указания расстояний l_{ij}, $i = \overline{1,7},\ j = \overline{i+1,8}$;
- ячейки D23:J23, E24:J24, F25:J25, G26:J26, H27:J27, I28:J28, J29 – для промежуточных результатов с загрузкой формулы для вычисления произведения $l_{ij}x_{ij}$, $i = \overline{1,7},\ j = \overline{i+1,8}$;
- ячейки D31:J31 – для промежуточных результатов с загрузкой формулы для вычисления произведения сумм $\sum_{i=1}^{j-1} l_{ij}x_{ij},\ \ j = \overline{2,8}$;
- ячейка L31 – для целевой функции y с соответствующей загруженной формулой для её вычисления.

Все ячейки электронной таблицы, задействованные в решении задачи должны иметь *числовой* формат с числом десятичных знаков, равным 0.

После загрузки в электронную таблицу всех необходимых констант и формул для вычисления промежуточных и конечного результатов следует выполнить необходимые установки данных в диалоговом окне команды *Сервис/Поиск решения*. В условиях примера установки определяются следующим образом:

- для целевой ячейки – **$L$31**, равной **минимальному значению;**
- для ячеек с изменяемыми переменными – **$D$4:$J$4**, **$C$5**, **$E$5:$J$5**, **$C$6:$D$6**, **$F$6:$J$6**, **$C$7:$E$7**,

$G$7:$J$7, **$C$8:$F$8**, **$H$8:$J$8**, **$C$9:$G$9**, **$I$9:$J$9**, **$C$10:$H$10**, **$J$10**, **$C$11:$I$11**;

- для ограничений: **$C$4:$J$11 = двоичное**;

$C$13: $J$13 = 1;

$L$4:$L$11 = 1;

$C$5 = $D$4;

$C$6 = $E$4;

. . .

$I$11 = $J$10.

$\left(2^n - n\right)$ равенств для каждой переменной x_{ij}, $i = \overline{1,8}$, $j = \overline{1,8}$, $i \neq j$.

На рис.1.26 показан вид экрана, который должен предшествовать выполнению команды *Сервис/Поиск решения*.

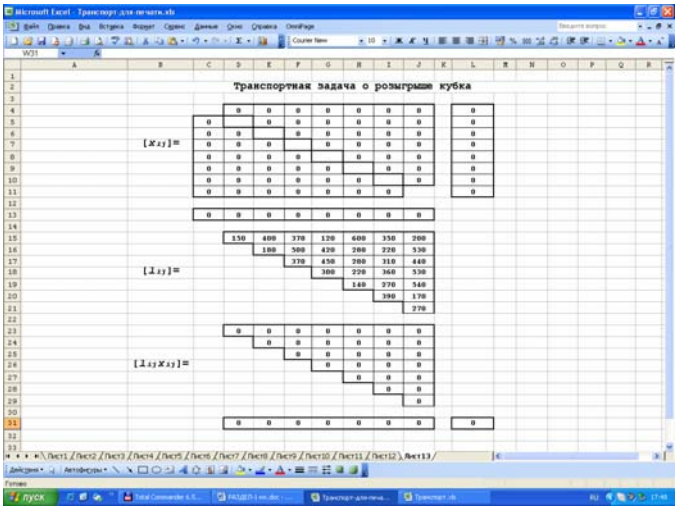


Рис. 1.26 – Экран с исходными установками

Вид экрана после выполнения команды *Сервис/Поиск решения* показан на рис. 1.27.

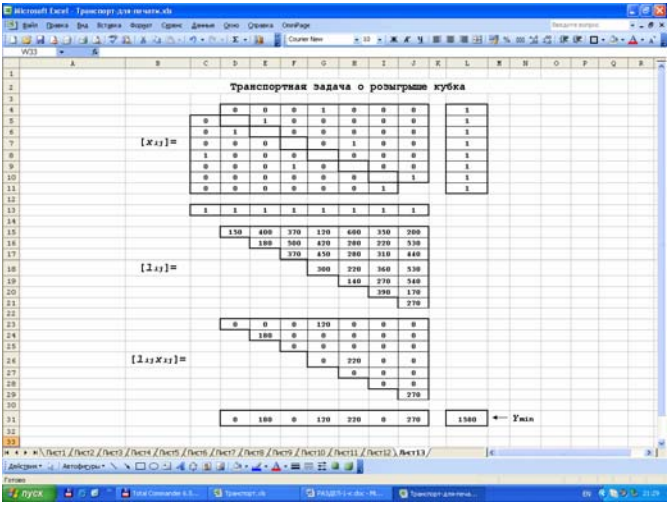


Рис. 1.27 – Вид экрана с промежуточными и конечными результатами

Оптимальным решением задачи являет матрица

$$\mathbf{X}^* = \begin{bmatrix} \times & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \times & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \times & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \times & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & \times & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \times & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \times & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \times \end{bmatrix},$$

которая обеспечивают минимальное значение целевой функции

$$y^* = 1540 \ \textit{км.}$$

Примечание. В матрице $\mathbf{X}^*$ символом «$\times$» обозначены элементы, которые в соответствии с требованием «$i \neq j$» не принимают участия в решении задачи.

Из решения оптимизационной задачи следует, что в очередном туре соревнований должны встретиться такие пары команд: 1-я и 5-я; 2-я и 3-я; 4-я и 6-я, 7-я и 8-я.

1.15. Задача о коммивояжере

Содержательная постановка задачи

Имеется ($n + 1$) город. Задана матрица $\mathbf{C} = [c_{\text{ij}}]$ – матрица расстояний между этими городами, $i, j = \overline{0, n}$. Выезжая из исходного города (ему приписывается номер 0), коммивояжер должен побывать во всех остальных городах ровно по одному разу и вернуться в город 0. В каком порядке следует объезжать города, чтобы пройденное суммарное расстояние было минимальным?

Математическая постановка задачи

Задачу коммивояжера можно сформулировать как задачу целочисленного линейного программирования. Введем двоичные переменные x_{ij}, $i = \overline{0, n}$, $j = \overline{0, n}$; $i \neq j$, имеющие следующий смысл:

$$x_{ij} = \begin{cases} 1, & \text{если коммивояжер из города } i \text{ переезжает в город } j, \\ 0 & \text{в противном случае.} \end{cases}$$

Введем также дополнительные переменные u_i, u_j $(i, j = \overline{1, n})$, принимающие произвольные вещественные значения (не умаляя общности, их можно считать целыми). Данные переменные позволят сформировать условие связности маршрута коммивояжера: исключить распадение маршрута на подциклы. Тогда математическая модель задачи имеет вид:

$$\sum_{i=0}^{n} \sum_{\substack{j=0 \\ j \neq i}}^{n} c_{ij} x_{ij} \to \min_{x_{ij} \in \Omega}, \tag{1.74}$$

$$\Omega: \quad \sum_{\substack{j=0 \\ j\neq i}}^{n} x_{ij} = 1, \quad i = \overline{0,n}\ , \tag{1.75}$$

$$\sum_{\substack{i=0 \\ i\neq j}}^{n} x_{ij} = 1, \quad j = \overline{0,n}\ , \tag{1.76}$$

$$u_i - u_j + nx_{ij} \leq n-1, \ \ i,j = \overline{1,n}, \tag{1.77}$$

$$u_i, u_j = \text{int}, \ \ j,i = \overline{1,n}. \tag{1.78}$$

$$x_{ij} + x_{ji} \leq 1, \ \ i,j = \overline{0,n}, \ \ i \neq j\ . \tag{1.79}$$

Здесь (1.74) определяет целевую функцию как суммарную длину маршрута коммивояжера. Условие (1.75) говорит о том, что коммивояжер въезжает в каждый город ровно один раз. Условие (1.76) говорит о том, что коммивояжер выезжает из каждого города ровно один раз. Ограничение (1.77) требует, чтобы любой маршрут коммивояжера состоял из одного цикла. Система равенств (1.78) ограничивает область допустимых значений дополнительных переменных целыми числами (положительными и отрицательными). Последнее ограничение исключает возврат коммивояжера в город, в котором он уже побывал.

Пример транспортной задачи о коммивояжере

Имеется пять городов 0, 1, 2, 3, 4 (рис. 1.28), и заданы расстояния между ними в виде матрицы

$$\mathbf{C} = [c_{ij}] = \begin{bmatrix} 0 & 12 & 4 & 9 & 10 \\ 12 & 0 & 11 & 6 & 8 \\ 4 & 11 & 0 & 13 & 5 \\ 9 & 6 & 13 & 0 & 7 \\ 10 & 8 & 5 & 7 & 0 \end{bmatrix}.$$

Выезжая из города 0, коммивояжер должен побывать во всех остальных городах ровно по одному разу и вернуться в город 0. В ка-

ком порядке следует объезжать города, чтобы пройденное суммарное расстояние было минимальным?

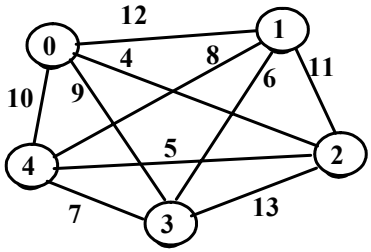


Рис. 1.28 – Условная схема городов с указанием расстояний между нами

Математическая модель транспортной задачи о коммивояжере

Математическая модель задачи при использовании обозначений, принятых для общей модели задачи коммивояжера (1.74) – (1.79), будет иметь вид:

$$y = 12x_{01} + 4x_{02} + 9x_{03} + 10x_{04} + 12x_{10} + 11x_{12} + 6x_{13} +$$
$$+ 8x_{14} + 4x_{20} + 11x_{21} + 13x_{23} + 5x_{24} + 9x_{30} + 6x_{31} +$$
$$+ 13x_{32} + 7x_{34} + 10x_{40} + 8x_{41} + 5x_{42} + 7x_{43} \to \min_{x_{ij} \in \mathbf{\Omega}},$$

$$\mathbf{\Omega}: \; f_0 = x_{01} + x_{02} + x_{03} + x_{04} = 1,$$
$$f_1 = x_{10} + x_{12} + x_{13} + x_{14} = 1,$$
$$f_2 = x_{20} + x_{21} + x_{23} + x_{24} = 1,$$
$$f_3 = x_{30} + x_{31} + x_{32} + x_{34} = 1,$$
$$f_4 = x_{40} + x_{41} + x_{42} + x_{43} = 1,$$
$$f_5 = x_{10} + x_{20} + x_{30} + x_{40} = 1,$$
$$f_6 = x_{01} + x_{21} + x_{31} + x_{41} = 1,$$
$$f_7 = x_{02} + x_{12} + x_{32} + x_{42} = 1,$$

$$f_8 = x_{03} + x_{13} + x_{23} + x_{43} = 1,$$
$$f_9 = x_{04} + x_{14} + x_{24} + x_{34} = 1,$$
$$u_i - u_j + 4x_{ij} \le 3, \quad i, j = \overline{1,4}, \ i \ne j,$$
$$u_i, u_i = \text{int}, \ i = \overline{1,4}.$$
$$x_{ij} + x_{ji} \le 1, \ i, j = \overline{0,4}, \ i \ne j.$$

Цифровая модель и решение задачи в информационной среде *Microsoft Excel*

Распределение и назначение ячеек электронной таблицы для транспортной задачи о коммивояжере в условиях примера могут быть следующими:

- ячейки C5:F5, B6, D6:F6, B7:C7, E7:F7, B8:D8, F8, B9:E9 – для искомых переменных задачи x_{ij}, $i, j = \overline{1,4}$;
- ячейки C11:F11 – для дополнительных переменных задачи u_j, $j = \overline{1,4}$;
- ячейки H5:H9 – для дополнительных переменных задачи u_i, $i = \overline{1,4}$;
- ячейки B14:F14 – для функций f_i, $i = \overline{0,4}$, с соответствующими загруженными формулами;
- ячейки J4:J9 – для функций f_{4+j}, $j = \overline{0,4}$, с соответствующими загруженными формулами;
- ячейки B16:F20 – для указания расстояний между городами l_{ij}, $i, j = \overline{0,4}$, $i \ne j$;
- N6:P6, O7:P7, M8:N8, P8, M9:O9 – для промежуточных результатов с загруженными формулами для вычисления выражения $u_i - u_j + 4x_{ij} \le 3$, $i, j = \overline{1,4}$, $i \ne j$,

- H11:K11, M7, I12:I12, J13:K13, K14 – для промежуточных результатов с загруженными формулами для вычисления выражения $x_{ij} + x_{ji},\ i, j = \overline{0,4},\ i \neq j$;
- H16:L20 – для промежуточных результатов с загруженными формулами для вычисления произведения $c_{ij} x_{ij}, i, j = \overline{0,4},\ i \neq j$;
- ячейка N18 – для целевой функции *y* с соответствующей загруженной формулой для её вычисления.

Все ячейки электронной таблицы, задействованные в решении задачи должны иметь *числовой* формат с числом десятичных знаков, равным 0.

После загрузки в электронную таблицу всех необходимых констант и формул для вычисления промежуточных и конечного результатов следует выполнить необходимые установки данных в диалоговом окне команды *Сервис/Поиск решения*. В условиях примера установки определяются следующим образом:

- для целевой ячейки – **$N$18**, равной **минимальному значению;**
- для ячеек с изменяемыми переменными – **$C$5:$F$5**;

  $B$6; **$D$6:$F$6**; **$B$7:$C$7**; **$E$7:$F$7**; **$B$8:$D$8**;

  $F$8; **$B$9:$E$9**; **$C$11:$F$11**; **$H$5:$H$9**;
- для ограничений: **$B$14:$F$14 = 1**,

  $J$5:$J$9 = 1,

  $C$5:$F$5 = двоичное,

  $B$6 = двоичное,

  $B$7:$C$7 = двоичное,

  $B$8:$D$8 = двоичное,

  $B$9:$E$9 = двоичное,

  $D$6:$F$6 = двоичное,

  $E$7:$F$7 = двоичное,

  $F$8 = двоичное,

  $B$11:$F$11 = целое,

  $H$5:$H$9 = целое,

$N$6:$P$6 <= 3,
$M$7 <= 3,
$O$7:$P$7 <=3,
$M$8:$N$8 <= 3,
$P$8 <= 3,
$M$9:$O$9 <= 3,
$H$11:$K$11 <= 1,
$I$12:$K$12 <= 1,
$J$13:$K$13 <= 1,
K$14 <= 1.

На рис.1.29 показан вид экрана, который должен предшествовать выполнению команды *Сервис/Поиск решения*.

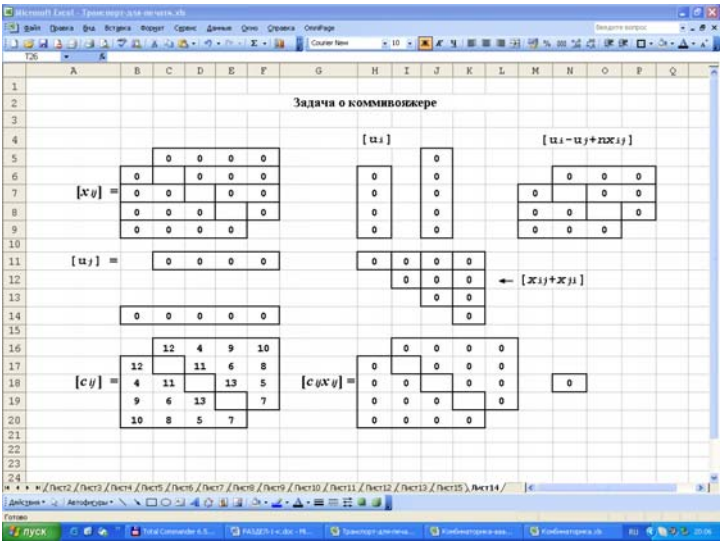


Рис. 1.29 – Вид экрана с исходными установками

Вид экрана после выполнения команды *Сервис/Поиск решения* показан на рис. 1.30.

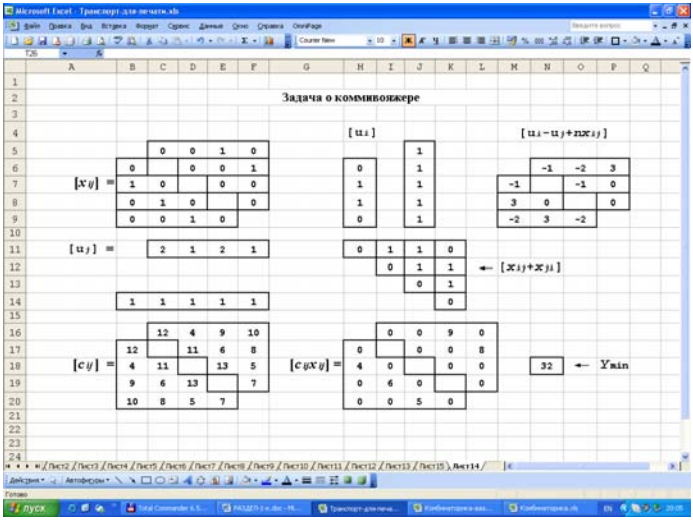


Рис. 1.30 – Вид экрана с промежуточными и конечными результатами

Решение задачи $\mathbf{X}^* = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$ определяет покрытие, соответствующее графу на рис. 1.28. При этом $y_{\min} = 32$.

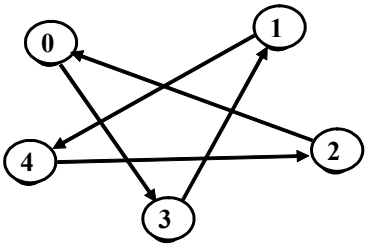


Рис. 1.31 – Условная схема городов с указанием оптимального маршрута коммивояжера

* * *

Рассмотренные математическое модели далеко не охватывает все возможные типы оптимизационных транспортных задач. Бесспорно, реальная практика перевозок будет предлагать новые задачи с более сложными математическими моделями. Классификация задач, приведенная на рис. 1.1, будет постоянно расширяться и совершенствоваться.

Основная задача, которую ставили перед собой авторы – это, с одной стороны продемонстрировать читателям простейшую методику оптимизации задач с помощью встроенного в информационную систему *Microsoft Excel* программного средства «*Поиск решения*». С другой стороны, показать на задачах малой размерности адекватность моделей транспортных задач, которая в большинстве случаев может быть проверена прямым перебором возможных решений.

По рассмотренным решениям конкретных транспортных задач малой размерности можно составить представление об объёме и сложности работ по решению тех же задач большой размерности. Несмотря на то, что математические модели у задач малой и большой размерностей совпадают, решение последних с помощью рассмотренной методики не всегда представляется возможным. Если оптимизационные задачи малой размерности можно решить простым перебором и в реальном масштабе времени, то при решении задач большой размерности факторы времени счёта, расхода динамической памяти ЭВМ или сложности подготовки исходных данных начинают играть существенную роль и могут оказаться причиной, не позволяющей получить решение. Одним из возможных выходов может быть разработка новых методов решений, которые нейтрализуют выше перечисленные факторы. При разработке новых методов иногда приходится идти на компромисс, например, пожертвовать точностью решения ради экономии времени счёта.

Дальнейшие разделы монографии будут посвящены вопросам разработки методов решения транспортных задач по перевозке мелкопартионных грузов (большой размерности), их программной реализации и сравнительному анализу с существующими методами.

РАЗДЕЛ 2

РАЗБИЕНИЕ ТРАНСПОРТНОЙ СЕТИ НА РАЙОНЫ

Организация доставки мелкопартионных грузов в крупных городах или обширных регионах с малонаселенными пунктами связана с транспортной сетью, объединяющей большое количество потребителей и поставщиков. Для таких сетей оптимизация процесса перевозок является сложной и трудоёмкой задачей.

С целью более эффективной организации перевозок транспортную сеть разбивают на районы. При разбиении транспортной сети на районы учитывается карта местности, расстояния между поставщиками и потребителями, время проезда определённых участков дороги, объем поставок.

В данном исследовании в качестве критерия оценки проведенного разбиения выбрана суммарная длина маршрутов коммивояжера по всем районам разбиения. Альтернативным критерием выступает суммарное время проезда по всем маршрутам.

2.1. Анализ подходов к решению задачи районирования

Анализ требований, предъявляемых к качеству выполнения услуги доставки, показывает, что наиболее важными являются такие показатели доставки как своевременность, надежность, суммарные материальные затраты. В связи с этим весьма актуальным является исследование методов, позволяющих эффективно спланировать процесс доставки и оперирующих с перечисленными показателями транспортных услуг.

Поставщикам приходится обслуживать значительное количество клиентов, в связи с чем оптимизация процесса доставки выполняется для транспортных сетей большой размерности. Это затрудняет применение некоторых методов планирования маршрутов,

ограничивает возможность построения гибкого графика в случае неравномерности спроса, приводит к потере клиентов из-за несвоевременности доставки.

Более эффективной организации перевозочного процесса можно достичь, применив зонирование адресов доставки, или разбиение обслуживаемой транспортной сети на районы. Согласно автору [7], имеющему большой опыт практической работы с организацией транспортных перевозок, разбиение на районы и дополнительная фильтрация заданий на перевозку по зонам доставки при комплектации рейсов позволит избежать неоправданно завышенного пробега транспортных средств при отсутствии алгоритмов оптимальной маршрутизации.

Вопрос районирования транспортных сетей на данный момент широкого развития не получил. Известно лишь несколько подходов к разбиению транспортной сети на районы. Один из них [9] предполагает использование в качестве границ районов естественных препятствий – рек, железнодорожных путей и т. д. Данный подход обусловлен предположениями о том, что объезд естественных препятствий влечет за собой значительное увеличение общего пробега. Предполагается, что пункты, разделенные такими препятствиями, находятся на значительном расстоянии. Однако в условиях города данный подход может не обеспечить возможности разбить транспортную сеть на необходимое количество районов. Кроме того, полученные районы могут существенно отличаться по величине и по количеству включенных в них пунктов доставки.

Тот же автор предлагает альтернативный вариант разбиения транспортной сети, при котором транспортные районы совпадают с существующими административными районами. При этом в качестве границ района могут быть также использованы центральные улицы, оживленное движение на которых может существенно увеличить время, затрачиваемое на доставку внутри одного района. В разбиении, полученном исходя из таких соображений, в разных районах могут оказаться расположенные близко друг к другу пункты, включение которых в один маршрут было бы выгодным.

Достаточно эффективным может оказаться разбиение на районы, выполняемое экспертом, с учетом взаимного расположения пунктов доставки, а также карты местности, в которой расположена транспортная сеть. Однако такое разбиение является сугубо субъ-

ективным, и его формализация весьма затруднительно, какой бы критерий при этом не использовался.

В [10] разбиении сети на районы применяется как часть решения задачи развозки, полученного при помощи модификации метода Литтла-Кэрола-Мурти. В указанном случае транспортная сеть динамически разбивается на два микрорайона, один из которых является рабочим, а другой – нет. Динамический подход обеспечивает более быструю работу метода, однако не позволяет получить разбиение, которое может быть использовано как базовое разбиение транспортной сети. Такой подход не дает возможности получения подсетей, которые в дальнейшем можно было бы рассматривать автономно.

В статье [6] изложен алгоритм решения оптимизационной задачи доставки мелкопартионных грузов в условиях крупного города методом локализации. Локализация, или, по сути, разбиение транспортной сети города на районы, рассматривается как один из этапов процесса организации транспортных перевозок. Решение задачи основано на распределении всех клиентов по группам по территориальному признаку путём использования процедуры кластерного анализа на базе метода нечетких *c*-средних. Реализовать метод *c*-средних предлагается при помощи одной из встроенных функций пакета MATHLAB.

Метод нечетких *c*-средних [11] предназначен для разбиения множества элементов на нечеткие подмножества. Для каждого из элементов x_i определяется функция принадлежности к *j*-му множеству u_{ij}, $0 \le u_{ij} \le 1$. Метод основан на минимизации целевой функции

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \left\| x_i - c_j \right\|^2, \quad 1 \le m \le \infty, \tag{2.1}$$

где *m* - произвольное действительное число больше 1, x_i – вектор данных, характеризующий положение *i*-го объекта; c_j – вектор данных, характеризующий положение центра *j*-го кластера, $\|*\|$ – квад-

ратичная норма для определения связи между центрами и другими элементами.

Корректировка функции принадлежности производится по формуле

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\left\| x_i - c_j \right\|}{\left\| x_i - c_k \right\|} \right)^{\frac{2}{m-1}}}. \qquad (2.2)$$

Данные о положении центра корректируются в соответствии с формулой

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m}. \qquad (2.3)$$

Итеративный процесс приближения к оптимальному разбиению прекращается при выполнении условия $\max_{i,j} \left\{ \left| u_{ij}^{(k+1)} - u_{ij}^{(k)} \right| \right\} < \varepsilon$, где ε - заданная точность решения.

Применительно к задаче локализации пунктов транспортной сети метод предполагает следующую последовательность действий [6]. На основании данных о взаимном расположении пунктов транспортной сети выбираются центры кластеров (или районов) транспортной сети. Данные представляются в виде произвольного количества параметров каждого пункта или же в виде рисунка-карты с отмеченными на ней точками-пунктами (матрицы исходных данных, отображающей положение каждого из пунктов на карте). Согласно [6] кластеризация реализуется двумя разными функциями в зависимости от типа представленных данных.

Каждая точка данных рассматривается как потенциальный центр кластера. Для всех точек вычисляется определённая мера,

характеризующая способность точки выступать в качестве центра кластера. Мера конкретной точки в обязательном порядке зависит от плотности других точек вокруг нее.

При заданных параметрах пунктов алгоритм состоит из двух повторяющихся шагов:

1. Выбора центра кластера – точки с наибольшей мерой плотности.

2. Удаления всех точек в окрестности центра полученного кластера и выбора центра следующего кластера. Радиус окрестности должен быть задан заранее.

Повторение происходит до тех пор, пока все точки не окажутся внутри окрестностей заданного радиуса. Оптимальное количество кластеров определяется программой в ходе разбиения.

При графическом задании транспортной сети количество кластеров должно быть указано в исходных данных к процедуре. При поиске решения анализируется целевая функция. Процесс поиска решения также является итеративным. Он прекращается при достижении заданного числа итераций или при условии, что разность значений целевой функции на текущей и на предшествовавшей ей итерации не превышает заданной точности, как это было сказано ранее.

Обе функции в качестве результата выдают координаты центров кластеров и значения функции принадлежности для каждой пары точка-кластер.

Достоинством предлагаемого способа получения районов является наличие соответствующего программного обеспечения. К тому же довольно удобным является получение результата в виде нечетких множеств. Данная особенность дает возможность распределения периферийных пунктов по районам, исходя из дополнительных условий.

В работе [6] алгоритм охарактеризован как быстродействующий. Однако авторами данной работы разбиение на районы производилось только на основании расположения пунктов транспортной сети на плоскости, каждый пункт описывался парой координат по осям *X* и *У*. Особенности транспортных связей между пунктами,

естественные препятствия и т.д. не были учтены. Пакет MATLAB позволяет учесть также и эти параметры, однако такая задача потребует дополнительных исследований и может существенно снизить быстродействие алгоритма. Недостатком использования готового программного пакета является сложность внесения дополнительных параметров и ограничений.

В источнике [12] метод кластеризации охарактеризован как метод с большим временными затратами, при котором каждый дополнительный элемент увеличивает время решения задачи на целый порядок.

В работе [13] задача разбиения транспортной сети на районы формулируется как задача целочисленного линейного программирования с целевой функцией

$$\sum_{k=1}^{r}\sum_{i=1}^{n}\sum_{j=1}^{n} c_{i,j} x_{i,k} x_{j,k} \to \min \tag{2.4}$$

при ограничениях

$$\begin{cases} \sum_{k=1}^{r} x_{i,k} = 1, \quad \forall i \in \{1,...,n\}; \\ \sum_{i=1}^{n} d_i x_{i,k} \le w, \quad \forall k \in \{1,...,r\}; \\ x_{i,k} \in \{0,1\}, \quad \forall i \in \{1,...,n\}, \quad \forall k \in \{1,...,r\}, \end{cases} \tag{2.5}$$

где r – общее количество подмножеств разбиения, равное количеству имеющихся транспортных средств; $c_{i,j}$ – расстояние между *i*-м и *j*-м клиентами; $x_{i,k}$ – булевы переменные ($x_{i,k} = 1$, если *i*-й клиент входит в *k*-е подмножество разбиения, и $x_{i,k} = 0$ – в противном случае); d_i – количество груза, которое требуется доставить

i-му клиенту; *w* – максимально допустимый суммарный вес заказов всех клиентов в данном подмножестве разбиения, равный грузоподъемности транспортного средства.

В данной постановке предполагается, что в результате решения задачи будет получено разбиение транспортной сети на районы с минимальной протяженностью маршрутов. При этом суммарный объем заказов клиентов внутри одного района не будет превышать заранее заданного числа w, равного грузоподъемности автомобиля. Достоинством предлагаемой формулировки задачи разбиения транспортной сети на районы является емкость математической модели, которая учитывает как пробег транспортного средства внутри каждого из районов, так и его грузоподъемность. Недостатком является сложность решения задачи целочисленного линейного программирования (ЦЛП) как таковой. Согласно исследованиям авторов [14] о времени, затрачиваемом на решение задачи ЦЛП при помощи специальных методов сложно судить заранее, очень часто оно не зависит от размерности задачи. За приемлемое время можно гарантированно решить только задачи сравнительно небольшой размерности, что не является актуальным.

Таким образом, разработанные на данный момент системы рекомендаций не гарантируют оптимального разбиения транспортной сети на районы. Существующие подходы либо не формализованы и не приспособлены для применения на ЭВМ, либо не приспособлены для учета всех существенных факторов, либо не приспособлены для решения задач большой размерности. Кроме того, не существует критерия, позволяющего оценить эффективность разбиения транспортной сети на районы. В связи с этим исследование задачи разбиения транспортной сети на районы является необходимым и перспективным.

2.2. Методика проведения исследования

Для разбиения транспортной сети на районы разработан специальный метод, который базируется на известном комбинаторном методе «ветвей и границ». Полученная модификация метода явилась результатом многовекторного исследования автора А.А.Кобец. В процессе исследования были рассмотрены различные приемлемые варианты модификации метода «ветвей и границ». Все вариан-

ты модификации прошли комплексное тестирование, целью которого было установить степень удовлетворения каждым вариантом выбранному критерию качества разбиения.

Для оценки качества получаемого разбиения была проведена серия вычислительных экспериментов. На начальном этапе различные варианты модификации были протестированы на задачах малой размерности. Разбиение проводилось для транспортных сетей, содержащих от 8 до 18 пунктов доставки, на 2 или 3 района. Качество полученного разбиения оценивалось по суммарной длине маршрутов, полученных в результате решения задачи коммивояжера для каждого из районов в отдельности. Результаты разбиения транспортной сети на районы по модифицированному методу «ветвей и границ» сравнивались с результатами, полученными методом прямого перебора всех возможных вариантов разбиения, то есть с оптимальными результатами.

Для исследования были использованы пункты доставки, расположенные случайным образом. Расстояние между ними определялись по прямой. Пары непосредственно связанных между собой пунктов доставки отбирались случайным образом.

Количество пунктов в районах разбиения строго фиксировалось, и было равно, соответственно, одной второй (в случае разбиения на два района) или одной трети (в случае разбиения на три района) от общего количество пунктов. Такое ограничение позволило уменьшить количество рассматриваемых вариантов и обеспечило возможность сравнивать результаты разбиения с оптимальным вариантом.

Значительно актуальней является задача получения районов, занимающих примерно одинаковые площади, или же районов, при объезде всех пунктов которых потребуется проехать примерно одинаковые расстояния. При такой постановке требований к размерам района полный перебор для отыскания оптимального варианта возможен только для задач с очень малой размерностью, которые не представляют особого интереса. Поэтому при анализе механизма перебора вначале рассматривались алгоритмы разбиения на районы с равным количеством пунктов доставки. Среди этих алгоритмов был отобран наиболее приемлемый вариант, на основании которого и был разработан окончательный вариант модифициро-

ванного метода «ветвей и границ» применительно к задаче районирования.

Предлагаемая модификация метода «ветвей и границ» позволяет формировать районы, которые незначительно отличаются друг от друга длиной маршрутов коммивояжера.

Далее была проведена проверка окончательного алгоритма разбиения для пунктов доставки, случайным образом нанесенных на карту района города. Размерность задачи – до 60 пунктов доставки. Для оценки расстояний между пунктами доставки были определены длины кратчайших маршрутов, соединяющих эти пункты. Полученные разбиения транспортной сети на районы модифицированным методом „ветвей и границ” были нанесены на карту, после чего оценивались на предмет практической реализуемости и целесообразности.

Метод разбиения транспортной сети на районы был подобран экспериментально. Для оценки качества полученных разбиений транспортных сетей на районы вычислялись суммы длин маршрутов коммивояжера для каждого из полученных районов. Чтобы облегчить оценку качества подобранных алгоритмов, сначала рассматривались транспортные сети с небольшим количеством пунктов. При этом на количество пунктов в районах полученного разбиения накладывалось ограничение. Это позволило сравнивать разбиения, получаемые при помощи подобранных алгоритмов, с оптимальными (с точки зрения выбранного критерия разбиениями) вариантами, определённых методом полного перебора.

Затем ограничение на количество пунктов в районах было заменено более актуальным требованием незначительного различия между длинами маршрутов коммивояжера в полученных районах. Качество полученных разбиений оценивалось визуально после их нанесения на карту.

2.3. Адаптация метода «ветвей и границ» к решению задачи разбиения транспортной сети на районы

В настоящем подразделе рассматриваются вопросы адаптации метода «ветвей и границ» для решения задачи разбиения транспортной сети на районы.

Метод «ветвей и границ» традиционно используется для решения задачи коммивояжера. Согласно данному методу строится дерево. Вершинами дерева являются подмножества всего множества маршрутов, включающие или не включающие определенные звенья транспортной сети.

Для обозначения подмножества, состоящего из одной ветви дерева, соединяющей вершины i и j, обычно используют запись (i, j), для обозначения подмножества, не включающего данное звено – запись $(\overline{i, j})$.

Каждой вершине ставится в соответствие точная нижняя граница длины маршрута, соответствующая данному подмножеству. Вершины выбираются таким образом, чтобы нижняя граница длин соответствующих маршрутов была минимальной, и цикл образовывали только те маршруты, которые проходят через все вершины транспортной сети. В построенном дереве выбирается поддерево с минимальной длиной маршрута коммивояжера.

Для построения дерева и оценки нижних границ длин маршрутов формируется приведенная матрица расстояний, и все вычисления ведутся не на основании исходной матрицы расстояний, а на базе приведенной матрицы.

Предлагается на основании приведенной матрицы строить несколько деревьев в количестве, равном заданному числу районов в разбиении. И далее, все пункты, включенные в вершины каждого из деревьев, относить к одному району.

Приведём содержательную постановку задачи разбиения транспортной сети на районы, включающей требование к соизмеримости длин маршрутов различных районов.

Для рассматриваемой транспортной сети, содержащей N пунктов приема товара, заданы расстояния между пунктами, имеющими транспортную связь. Необходимо разбить сеть на R районов с минимальной суммарной длиной кольцевых маршрутов, получаемых при решении задачи коммивояжера для каждого из районов. При этом длина маршрутов каждого из районов не должна существенно отличаться от длин маршрутов остальных районов.

Формирование районов с помощью предлагаемого метода (модифицированного метода «ветвей и границ») условно разобьем на несколько этапов.

На первом этапе исходные пространственные данные транспортной сети преобразуются к виду, удобному для последующего компьютерного решения задачи районирования. При этом сначала составляется матрица расстояний $\mathbf{D}$ размерности $N \times N$, где *N* – число пунктов. Элемент d_{ij} данной матрицы равен либо длине пути из пункта *i* в пункт *j* в случае, если пункты *i* и *j* имеют непосредственную транспортную связь между собой ($i, j \in \{1,2,...,N\},\ i \neq j$), либо длине минимального маршрута из пункта *i* в пункт *j* – в противном случае. Длины маршрутов определяются при помощи матричного метода решения задачи о кратчайших расстояниях, описанного в [5].

Суть матричного метода состоит в следующем. На первом шаге задается матрица $\mathbf{R}^{(1)} = \left\{r_{ik}^{(1)}\right\}_{N\times N}$ с элементами

$$r_{ik}^{(1)} = \begin{cases} c(i,j), & (i,j) \in \mathbf{S}; \\ M, & (i,j) \notin \mathbf{S}, \end{cases}$$

где **S** – множество путей транспортной сети.

Далее для матриц $\mathbf{R}^{(2)}, \mathbf{R}^{(3)}, ..., \mathbf{R}^{(t+l)}, ..., \mathbf{R}^{(N)}$ определяются элементы

$$r_{ik}^{(t+l)} = \min\left\{ r_{ik}^{(t)},\ r_{ik}^{(l)},\ \min_{j}\left(r_{ij}^{(t)} + r_{jk}^{(l)}\right)\right\}.$$

Вычисления следует прекратить, если матрица с последующим номером совпадает с предыдущей.

Заканчивается этап преобразованием матрицы расстояний $\mathbf{D}$ в приведенную матрицу $\mathbf{D}''$.

Матрица $\mathbf{D}''$ формируется по формулам метода «ветвей и границ». Так, элементы матрицы $\mathbf{D}''$, согласно методу «ветвей и границ» определяются выражением

$$d''_{ij} = d'_{ij} - \min_i d'_{ij},\ i, j = \overline{1, N}, \tag{2.6}$$

где

$$d'_{ij} = d_{ij} - \min_j d_{ij}, \quad i, j = \overline{1, N}. \tag{2.7}$$

Смысл чисел матрицы $\mathbf{D}''$ следующий: пары пунктов, которым в данной матрице соответствуют нули, согласно методу «ветвей и границ», рекомендуется включить в кольцевой маршрут, проходящий через все вершины транспортной сети [5].

Очевидно, что разбиение сети на районы будет эффективным, если в один район включать пары пунктов, которым соответствуют наименьшие числа в приведенной матрице $\mathbf{D}''$.

На втором этапе начинается формирование первого района транспортной сети. Первый пункт формируемого района имеет номер $j_{1,1}$. Полученное разбиение зависит от этого пункта. Процедура выбора пункта $j_{1,1}$ начинается с вычисления характеристики средней удаленности каждого пункта от остальных пунктов транспортной сети по формуле

$$L_{cp.}(j) = \frac{1}{2(N-1)}\left(\sum_{\substack{i=1\\ i\neq j}}^{N} d_{ij} + \sum_{\substack{i=1\\ i\neq j}}^{N} d_{ji}\right), \quad j = \overline{1, N}. \tag{2.8}$$

Для получения наилучшего результата при помощи данного алгоритма следовало бы рассмотреть *N* возможных разбиений «Разб. 1», «Разб. 2», ..., «Разб. *N*», выбирая в качестве $j_{1,1}$ поочередно все пункты транспортной сети: $j_{1,1-P.1} = 1$, $j_{1,1-P.2} = 2$, ... Однако практические исследования показывают, что чаще всего

оптимальным оказывается разбиение, для которого у пункта $j_{1,1}$ оценка $L_{cp.}\left(j_{1,1}\right)$ является одной из самых больших. В зависимости от размеров транспортной сети и возможностей компьютера, на котором выполняется программа, предлагается рассматривать только *K* из *N* разбиений. Здесь $K = \frac{N}{2}$ для транспортных сетей с общим количеством пунктов $N \le 50$, $K = \frac{N}{3}$ для транспортных сетей, для которых $50 < N \le 100$, $K = 5 \div 15$ при $N > 100$.

Выбираемые первые пункты первых районов получаемых разбиений должны соответствовать условию

$$L_{cp.}\left(j_{1,1-P.1}\right) > L_{cp.}\left(j_{1,1-P..2}\right) > L_{cp.}\left(j_{1,1-P.3}\right) > \ldots$$

То есть для первого варианта разбиения первый пункт $j_{1,1}$ выбирается, исходя из условия

$$L_{cp.}\left(j_{1,1}\right) = \max_{j \in \mathbf{J}} L_{cp.}\left(j\right), \tag{2.9}$$

где **J** – множество пунктов, которые еще не были распределены по районам, $\mathbf{J} = \{1,2,\ldots,N\}$. А для разбиений 2, 3, …, k, …, K – исходя из условия

$$L_{cp.}\left(j_{1,1}\right) = \max_{j \in \mathbf{J}/\{j_1, j_2, \ldots, j_{k-1}\}} L_{cp.}\left(j\right), \tag{2.10}$$

где j_1 совпадает с номером $j_{1,1}$, выбранном при формировании разбиения *Разб.* 1, $L_{cp.}\left(j_{1,1}\right) = \max_{j \in \mathbf{J}} L_{cp.}\left(j\right)$; j_2 – с номером пункта $j_{1,1}$, выбранном при формировании разбиения *Разб.* 2, $L_{cp.}\left(j_{1,1} = j_2\right) = \max_{j \in \mathbf{J}/\{j_1\}} L_{cp.}\left(j\right)$ и т.д.

Затем к первому району присоединяются пункты $j_{2,1}, j_{3,1}, \ldots, j_{M,1}$. Порядок присоединения остальных пунктов от выбора первого пункта $j_{1,1}$ не зависит. Количество присоединяемых на данном этапе пунктов определяется выражением

$$M = \left[q \frac{N}{R} \right].$$

Здесь коэффициент *q* подбирается экспериментально. При решении задач малой размерности, до 20 пунктов, лучшие результаты получаются при $q = \frac{1}{2}$. Для задач большой размерности более приемлемым является $q = \frac{1}{3}$.

Для присоединения к формируемому району пункта $j_{s,1}$, $s \in \left\{ \overline{2, M} \right\}$, значение критерия $E_{j_{s,1}}$ должно удовлетворять условию

$$E_{j_{s,1}} = \min_{\substack{j=\overline{1,N} \\ j \neq j_{t,1},\ t=\overline{1,s-1}}} \left\{ E_j \right\}. \qquad (2.11)$$

Критерий E_j вычисляется в соответствии с формулой:

$$E_j = \min \left\{ \min_{i \in \{ j_{1,1}, j_{2,1}, \ldots, j_{s-1,1} \}} d''_{ij} \ , \ \min_{i \in \{ j_{1,1}, j_{2,1}, \ldots, j_{s-1,1} \}} d''_{ji} \right\}. \qquad (2.12)$$

На следующем этапе начинается формирование остальных районов. Для *r*-го района, $r \in \{2, R\}$, номер начального пункта $j_{1,r}$ определяется, исходя из выполнения условия

$$L'_{cp.}\left(j_{1,r}\right)=\max_{j\in\mathbf{J}} L'_{cp.}\left(j\right), \tag{2.13}$$

для характеристики $L'_{cp.}\left(j\right)$, вычисляемой по формуле

$$L'_{cp.}\left(j\right)=\frac{1}{2M\left(r-1\right)}\left(\sum_{\substack{i\in\mathbf{I}_t\\ t=1,\ldots,r-1}} d_{ij}+\sum_{\substack{i\in\mathbf{I}_t\\ t=1,\ldots,r-1}} d_{ji}\right),\quad j\in\mathbf{J}\,. \tag{2.14}$$

К каждому из районов присоединяется $M-1$ пункт. Критерий присоединения пункта $j_{s,r}$, $s\in\{2,3,\ldots,M\}$, аналогичен критерию, который применялся на предыдущем этапе. Пункт $j_{s,r}$ выбирается из условий:

$$E_{j_{s,r}}=\min_{j\in\mathbf{J}}\{E_j\}, \tag{2.15}$$

$$E_j=\min\left\{\min_{i\in\mathbf{I}_r} d''_{ij},\ \min_{i\in\mathbf{I}_r} d''_{ji}\right\}, \tag{2.16}$$

где I_r – множество пунктов, присоединенных к району *r*.

Таким образом, мы получаем *R* районов, каждый из которых включает в себя *M* пунктов. Кроме того, имеется $\left(N-MR\right)$ пунктов, которые не были отнесены ни к одному из районов.

Распределение оставшихся $N-MR$ пунктов по районам осуществляется следующим образом.

Для каждого района определяем длину кольцевого маршрута, проходящего через все пункты. Вычисляем среднюю длину полученных маршрутов и накладываем запрет на добавление новых пунктов к тем районам, «размер» которых превышает среднее зна-

чение более чем на *p* процентов. Число *p* задается в зависимости от допустимого различия между размерами районов.

На практике получить районы с одинаковыми «размерами», то есть с одинаковой длиной маршрута коммивояжера, или с размерами, различающимися на заданное количество процентов, возможно далеко не всегда. Причина этого – дискретность определяемых таким образом «размеров» районов. Кроме того, оценка длины маршрута коммивояжера является достаточно трудоемкой задачей для районов с большим количеством пунктов. Поэтому тщательный «контроль» размеров районов может привести к чрезмерному количеству необходимых вычислений. При этом существенно сократятся размеры транспортных сетей, которые можно разбить на районы, используя данный алгоритм.

Таким образом, на практике параметр *p* будет влиять на различие между размерами районов, но фактическое различие может от него отличаться.

Для очередного района, на добавление пунктов к которому не был введен запрет, определяем наиболее выгодные для присоединения элементы. Пусть в текущем районе *r* уже содержится s пунктов. Тогда, в случае присоединения очередного пункта к району, количество пунктов в районе станет равным $s+1$. Для определения номера этого пункта используем критерий E'_j, вычисляемый в соответствии с формулами:

$$E'_j = \min\left\{\begin{array}{l} \min\left\{ d''_{i1j} + d''_{i2j} \middle| d''_{i1j}, d''_{i2j} \le \\ \qquad \le d''_{ij},\, i, i1, i2 \in \mathbf{I}_r,\, i \ne i1,\, i \ne i2 \right\}, \\ \min\left\{ d''_{ji1} + d''_{ji2} \middle| d''_{ji1}, d''_{ji2} \le \\ \qquad \le d''_{ji},\, i, i1, i2 \in \mathbf{I}_r,\, i \ne i1,\, i \ne i2 \right\} \end{array}\right\}, \quad (2.17)$$

если $s+1 \in \{[qN/R]+1, \quad [2qN/R]\}$;

$$E'_j = \min\left\{\begin{array}{l} \min\left\{\begin{array}{l} d''_{i1j} + d''_{i2j} + d''_{i3j} \left| d''_{i1j}, d''_{i2j}, d''_{i3j} \le d''_{ij}, \\ i, i1, i2, i3 \in \mathbf{I}_r, i \ne i1, i \ne i2, i \ne i3 \end{array}\right\}, \\ \min\left\{\begin{array}{l} d''_{ji1} + d''_{ji2} + d''_{ji3} \left| d''_{ji1}, d''_{ji2}, d''_{ji3} \le d''_{ji}, \\ i, i1, i2, i3 \in \mathbf{I}_r, i \ne i1, i \ne i2, i \ne i3 \end{array}\right\} \end{array}\right\}, \quad (2.18)$$

если $s+1 \in \{[2qN/R]+1, [N/R]\}$.

Как и в случае с критерием E_j, для выбора пункта $j_{s+1,r}$ необходимо выполнение условия

$$E'_{j_{s+1,r}} = \min_{j \in \mathbf{J}} \{E'_j\}. \quad (2.19)$$

Пусть для присоединения к району *r* в соответствии с указанным выше критерием был отобран пункт j_m ($j_m = j_{s+1,r}$). Для этого пункта определим «ближайший» район. «Ближайшим» будем считать район r_k, для которого справедливы соотношения:

$$E''_{r_k, j_m} = \min_{r_t \in \{1,\ldots R\}} \{E''_{r_t, j_m}\}, \quad (2.20)$$

$$E''_{r_t, j_m} = \min\left\{\begin{array}{l} \min\left\{\begin{array}{l} d''_{i1j_m} + d''_{i2j_m} \left| d''_{i1j_m}, d''_{i2j_m} \le d''_{ij_m}, \\ i \in \mathbf{I}_r,\ i \ne i1,\ i \ne i2 \end{array}\right\} \\ \min\left\{\begin{array}{l} d''_{j_m i1} + d''_{j_m i2} \left| d''_{j_m i1}, d''_{j_m i2} \le d''_{j_m i}, \\ i \in \mathbf{I}_r,\ \ i \ne i1,\ i \ne i2 \end{array}\right\} \end{array}\right\}. \quad (2.21)$$

Если «ближайшим» районом r_k является формируемый район *r,* то пункт j_m присоединяется к району *r.*

Повторяем поиск «ближайшего пункта» для следующего района, к которому можно присоединять пункты. Снова выполняем проверку, является ли район «ближайшим» для выбранного пункта, и т.д.

Процесс завершается, когда множество $\mathbf{J}$ пунктов, не присоединенных ни к одному из районов, становится пустым.

2.3.1. Алгоритм модифицированного метода «ветвей и границ»

Ниже приведен алгоритм предлагаемой модификации метода «ветвей и границ», позволяющий более точно представить предлагаемый порядок действий.

Шаг 1. Производится ввод расстояний между пунктами транспортной сети (только для тех пар пунктов, которые связаны между собой непосредственно, без транзитных пунктов).

Шаг 2. Формируется матрица расстояний между пунктами $\mathbf{D}$. Расстояние в матрице $\mathbf{D}$ – длина пути между пунктами, либо длина кратчайшего маршрута, соединяющего пункты.

Шаг 3. Для всех пунктов $j = \overline{1, N}$ вычисляются оценки среднего расстояния до пунктов $L_{cp.}(j)$ по формуле (2.8).

Шаг 4. Вычисляется приведенная матрица $\mathbf{D}''$ по формулам (2.6), (2.7).

Шаг 5. Номер варианта разбиения, который будет получен, приравнивается к единице, $k := 1$.

Шаг 6. Номер района, который будет сформирован, приравнивается к единице, $r := 1$. Формируется множество $\mathbf{J}$ пунктов, которые еще не были присоединены ни к одному из районов. В начале реализации алгоритма к ним относятся все пункты транспортной

сети, то есть $\mathbf{J} := \{1,2,...,N\}$. Задаются множества пунктов $\mathbf{I}_1$, $\mathbf{I}_2$, …, $\mathbf{I}_R$ – множества пунктов соответственно в районах 1, 2, …, R. Поскольку к ним еще не был присоединен ни один пункт, то $\mathbf{I}_1 = \mathbf{I}_2 = ... = \mathbf{I}_R = \varnothing$.

Шаг 7. Задается начальное значение счётчика пунктов, присоединенных к району, $s := 1$.

Шаг 8. Определяется первый пункт первого района $j_{1,1}$, исходя из условия (2.9), если $k = 1$, или из условия (2.10), если $k > 1$. Выбранный пункт $j_{1,1}$ перемещается из множества $\mathbf{J}$ в множество $\mathbf{I}_1$. С этого момента количество пунктов в множестве $\mathbf{I}_1$ становится равным единице, то есть теперь заданный номер пункта *s*, который присоединяется к району, можно рассматривать также как мощность множества пунктов района, $s = card\,\mathbf{I}_1$.

Шаг 9. Номер текущего пункта района *s* увеличивается на единицу, $s := s + 1$.

Шаг 10. Осуществляется выбор пункта $j_{s,1}$ для присоединения к формируемому первому району. Выбор производится, исходя из выполнения условия (2.11) для критерия E_j, вычисленного по формуле (2.12). Выбранный пункт $j_{s,1}$ перемещается из множества $\mathbf{J}$ в множество $\mathbf{I}_1$.

Шаг 11. Если количество пунктов в первом районе $card\,\mathbf{I}_1 = s < M$, то происходит возврат к шагу 9.

Шаг 12. Номер текущего района увеличивается на единицу, $r := r + 1$.

Шаг 13. Номеру присоединяемого к району пункта *s* присваивается единица, $s := 1$.

Шаг 14. Выбирается первый пункт для присоединения к текущему району $j_{1,r}$, в соответствии с условием (2.13) для характеристики $L'_{ср.}(j)$, вычисленной по формуле (2.14). Выбранный пункт $j_{1,r}$ перемещается из множества $\mathbf{J}$ в множество $\mathbf{I}_r$.

Шаг 15. Номер текущего пункта в районе увеличивается на единицу, $s := s+1$.

Шаг 16. Выбирается пункт $j_{s,r}$, исходя из выполнения условия (2.15) для критерия E_j, вычисляемого по формуле (2.16). Пункт $j_{s,r}$ перемещается из множества $\mathbf{J}$ в множество $\mathbf{I}_r$.

Шаг 17. Если количество пунктов в текущем районе $card\,\mathbf{I}_r = s < M$, то осуществляется возврат к шагу 15.

Шаг 18. Номер текущего района *r* приравнивается к единице, $r := 1$.

Шаг 19. Определяется «размер» каждого из районов $Size(r)$, $r = \overline{1,R}$, то есть длина маршрута коммивояжера для каждого из полученных районов, а также среднее арифметическое $\overline{Size}$ размеров всех полученных районов по формуле

$$\overline{Size} = \frac{1}{R}\sum_{r=1}^{R} Size(r).$$

Данные вычисления производятся только при условии, что со времени предыдущего вычисления «размеров» районов происходили изменения в составах множеств $\mathbf{J}$ и $\mathbf{I}_r$, $r = \overline{1,R}$. Чтобы ускорить быстродействие алгоритма рекомендуется определять «размеры» районов после присоединения нескольких пунктов к районам.

Шаг 20. Для текущего района проверяется выполнение условия

$$Size(r) \le \overline{Size} \cdot (p+1).$$

В случае, если это условие не выполняется, происходит переход к шагу 25.

Шаг 21. Осуществляется выбор пункта $j_{s+1,r}$ исходя из выполнения условия (2.19) для критерия E'_j, вычисленного по формуле (2.17) или (2.18) в зависимости от числа $s+1$. Здесь *s* - количество пунктов в текущем районе, $s = card\, I_r$.

Шаг 22. Для пункта $j_{s+1,r}$ осуществляется выбор «ближайшего» района r_k, исходя из выполнения условия (2.20) для критерия $E''_{r_t, j_{s+1,r}}$, вычисляемого по формуле (2.21).

Шаг 23. Если номер «ближайшего» к пункту $j_{s+1,r}$ района r_k и номер формируемого района *r* не совпадают, $r_k \ne r$, то происходит переход к шагу 25.

Шаг 24. Пункт $j_{s+1,r}$ перемещается из множества $\mathbf{J}$ в множество $\mathbf{I}_r$.

Шаг 25. Если номер текущего района $r < R$, то реализуется следующий шаг. В противном случае происходит переход к шагу 27.

Шаг 26. Номер текущего района увеличивается на единицу, $r := r+1$. Осуществляется переход к шагу 28.

Шаг 27. Номеру текущего района присваивается единица, $r := 1$.

Шаг 28. Если множество $\mathbf{J}$ не пустое, $\mathbf{J} \ne \varnothing$, то осуществляется возврат к шагу 19.

Шаг 29. Вычисляется оценка полученного разбиения $Eval._{cur.}$, то есть нахождение суммарной длины маршрутов коммивояжера всех районов, $Eval._{cur.} = \sum_{r=1}^{R} Size(r)$.

Шаг 30. Если был найден первый вариант разбиения, $k = 1$, то происходит переход к шагу 32.

Шаг 31. Если оценка полученного разбиения $Eval._{cur.}$ больше или же совпадает с наилучшей из полученных оценок $Eval._{\min.}$, $Eval._{cur.} \geq Eval._{\min}$, то происходит переход к шагу 33.

Шаг 32. Оценка полученного разбиения сохраняется как наилучшая из оценок, $Eval._{\min.} := Eval._{cur.}$. Полученное распределение пунктов по районам сохраняется как наилучшее из разбиений.

Шаг 33. Номер варианта разбиения увеличивается на единицу, $k := k + 1$.

Шаг 34. Если еще не превышено заданное количество разбиений, то есть $k \leq K$, то происходит возврат к шагу 6.

Шаг 35. На экран выводится распределение пунктов по районам, соответствующее наилучшему из разбиений, то есть множества $\mathbf{I}_1, \mathbf{I}_2, ..., \mathbf{I}_R$, и оценка наилучшего разбиения $Eval._{\min}$.

Последовательность действий при разбиении транспортной сети на районы с помощью модифицированного метода «ветвей и границ» можно представить в виде блок-схемы, представленной на рис. 2.1 – 2.2.

При разработке модифицированного метода «ветвей и границ» к решению задачи разбиения на районы изначально на количество пунктов в транспортной сети было наложено требование: общее количество пунктов *N* должно быть кратно количеству районов, то есть

$$N = kR, \quad k \in \{2,3,...\}. \tag{2.22}$$

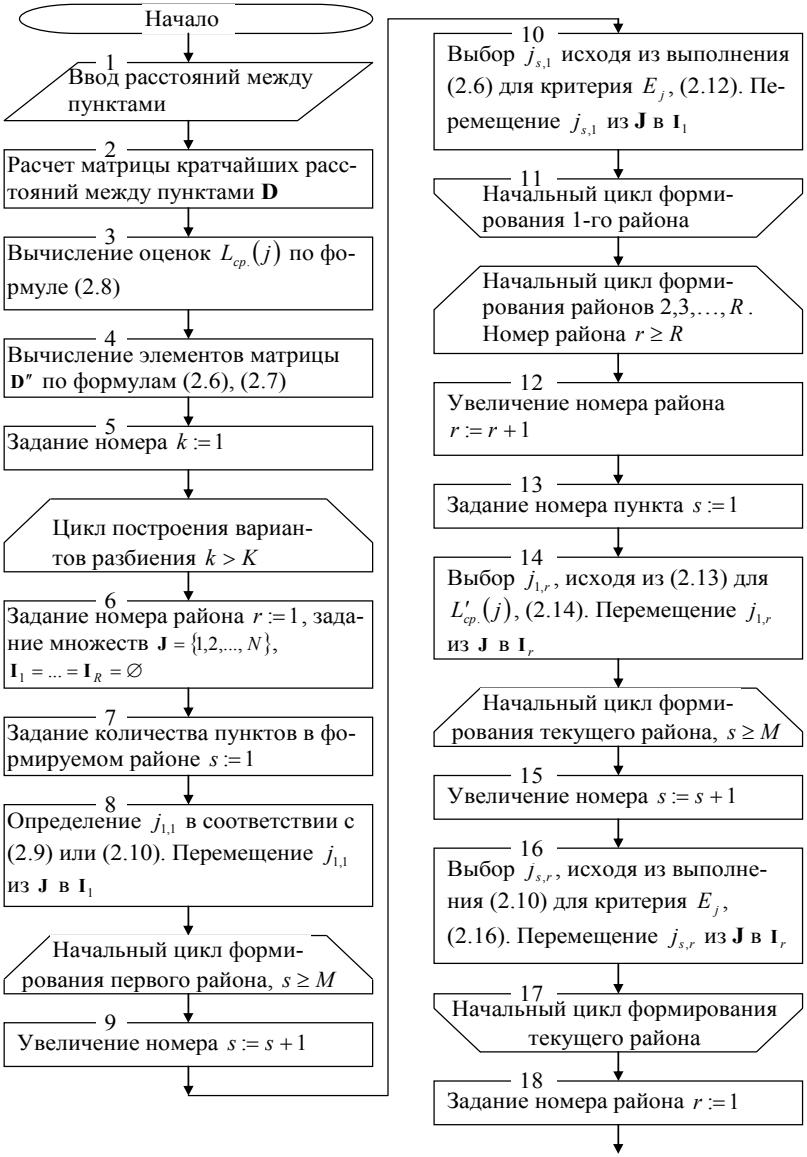


Рис. 2.1 – Начало схемы алгоритма модифицированного метода «ветвей и границ»

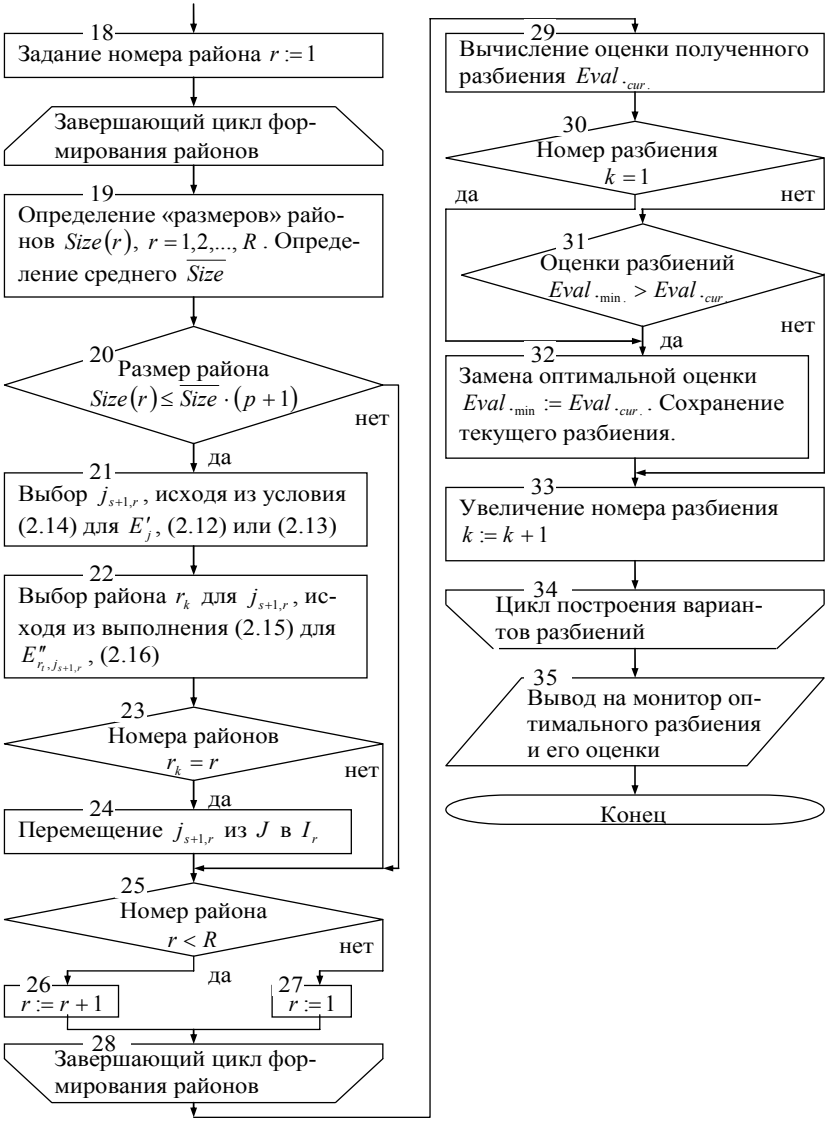


Рис. 2.2 – Окончание схемы алгоритма модифицированного метода «ветвей и границ»

Требование (2.22) гарантирует равенство количеств пунктов во всех районах разбиения $card\,\mathbf{I}_r = \dfrac{N}{R} = k,\ \ r = \overline{1,R}$.

Эффективность протестированных алгоритмов решения задачи районирования оценивалась при помощи сравнения результатов разбиения с оптимальным результатом, полученным при помощи полного перебора всех возможных вариантов разбиения на районы. Сравнение производилось только для вариантов разбиения, отвечающих условию (2.22).

Ниже описаны все алгоритмы приемлемых модификаций метода «ветвей и границ», которые были протестированы в равных условиях – для одних и тех же транспортных сетей.

2.3.2. Первый вариант алгоритма модифицированного метода «ветвей и границ»

Шаг 1. Вводится матрица расстояний $\mathbf{D}$.

Шаг 2. В соответствии с формулами (2.6), (2.7) формируется приведенная матрица $\mathbf{D}''$ и вектор $\tilde{\mathbf{d}}$, элементы которого

$$\tilde{d}_j = \frac{\sum\limits_{\substack{i=1 \\ i \neq j}}^{N} d_{ij}}{N},\ \ j = \overline{1,N}. \qquad (2.23)$$

Шаг 3. Устанавливается номер текущего района $r := 1$, формируются множество пунктов, не присоединенных ни к одному из районов $\mathbf{J} = \{1,2,...,N\}$, и множества пунктов, входящих в районы, $\mathbf{I}_1 = \mathbf{I}_2 = ... = \mathbf{I}_R = \varnothing$.

Шаг 4. Номеру пункта текущего района присваивается единица, $s := 1$.

Шаг 5. Выбирается первый элемент $j_{1,r}$ района *r* в соответствии с условием

$$\widetilde{d}_{j_{1,r}} = \max_{j \in \mathbf{J}} \widetilde{d}_j . \qquad (2.24)$$

Элемент $j_{1,r}$ перемещается из множества $\mathbf{J}$ в множество элементов района $\mathbf{I}_r$.

Шаг 6. Количество элементов в районе увеличивается на единицу, $s := s + 1$.

Шаг 7. Исходя из условия (2.15) для критерия E_j, вычисляемого по формуле (2.16), выбирается элемент $j_{s,r}$. Выбранный пункт перемещается из множества $\mathbf{J}$ в множество $\mathbf{I}_r$.

Шаг 8. Если количество элементов в районе $s < \dfrac{N}{R}$, то осуществляется переход на шаг 6.

Шаг 9. Если $r < R$, то номер района увеличивается на единицу: $r := r + 1$, и осуществляется возврат к шагу 5.

Шаг 10. Полученный результат выводится на экран. Он представляет собой списки пунктов, которые следует отнести к каждому из районов. Также выводятся длины маршрутов коммивояжера для каждого из районов рекомендуемого разбиения и суммарная длина маршрутов для всех районов.

Блок-схема алгоритма 1 приведена на рис. 2.3.

2.3.3. Второй вариант алгоритма модифицированного метода «ветвей и границ»

Шаг 1. Вводится матрица расстояний $\mathbf{D}$.

Шаг 2. В соответствии с формулами (2.6), (2.7) формируется приведенная матрица $\mathbf{D}''$. Формируется вектор $\widetilde{\mathbf{d}}$, элементы которого вычисляются по формуле (2.23).

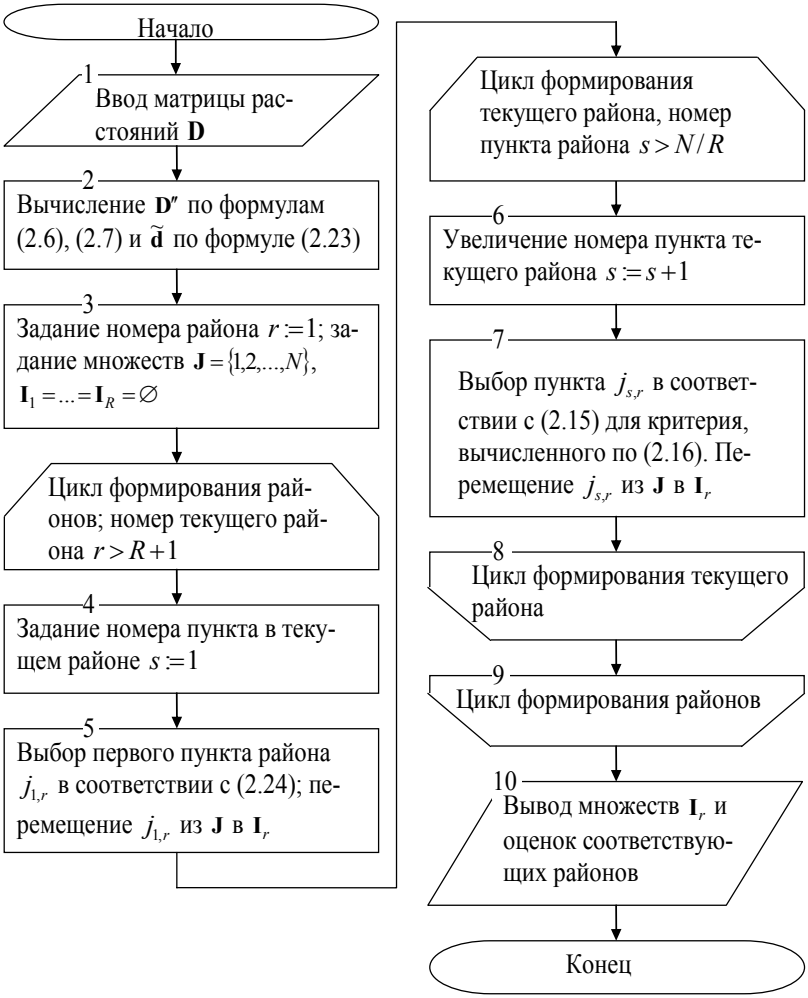


Рис. 2.3 − Схема алгоритма 1

Шаг 3. Задается множество пунктов, не присоединенных ни к одному из районов $\mathbf{J} = \{1,2,..., N\}$, множества пунктов, входящих в

районы, $\mathbf{I}_1 = \mathbf{I}_2 = ... = \mathbf{I}_R = \varnothing$. Номер текущего пункта каждого из районов s приравнивается к единице, $s := 1$.

Шаг 4. Выбираются первые элементы $j_{1,r}$ для всех районов r, $r = \overline{1,R}$, в соответствии с условием (2.24).

Элементы $j_{1,r}$ перемещаются из множества **J** в соответствующие множества элементов районов $\mathbf{I}_r$, $r = \overline{1,R}$.

Шаг 5. Номер текущего пункта всех районов увеличивается на 1, $s := s + 1$.

Шаг 6. Если количество пунктов в районах $s = \dfrac{N}{R} + 1$, то разбиение считается завершенным, иначе осуществляется переход к шагу 11.

Шаг 7. Номеру текущего района присваивается единица, $r := 1$.

Шаг 8. К множеству пунктов текущего района $\mathbf{I}_r$ добавляется пункт $j_{s,r}$. Для пункта $j_{s,r}$, добавляемого к *r*-му району, $r = \overline{1,R}$, обязательным является выполнение условия (2.15) для критерия E_j, вычисляемого по формуле (2.16).

Шаг 9. Номер текущего района увеличивается на 1, $r := r + 1$.

Шаг 10. Если номер текущего района превышает R, $r > R$, то осуществляется переход к шагу 5. В противном случае – переход к шагу 8.

Шаг 11. На экран выводятся списки пунктов, которые рекомендуется отнести к каждому из районов, а также длины маршрутов коммивояжера для каждого из этих районов и их суммарная длина.

Блок-схема алгоритма 2 приводится на рис. 2.4.

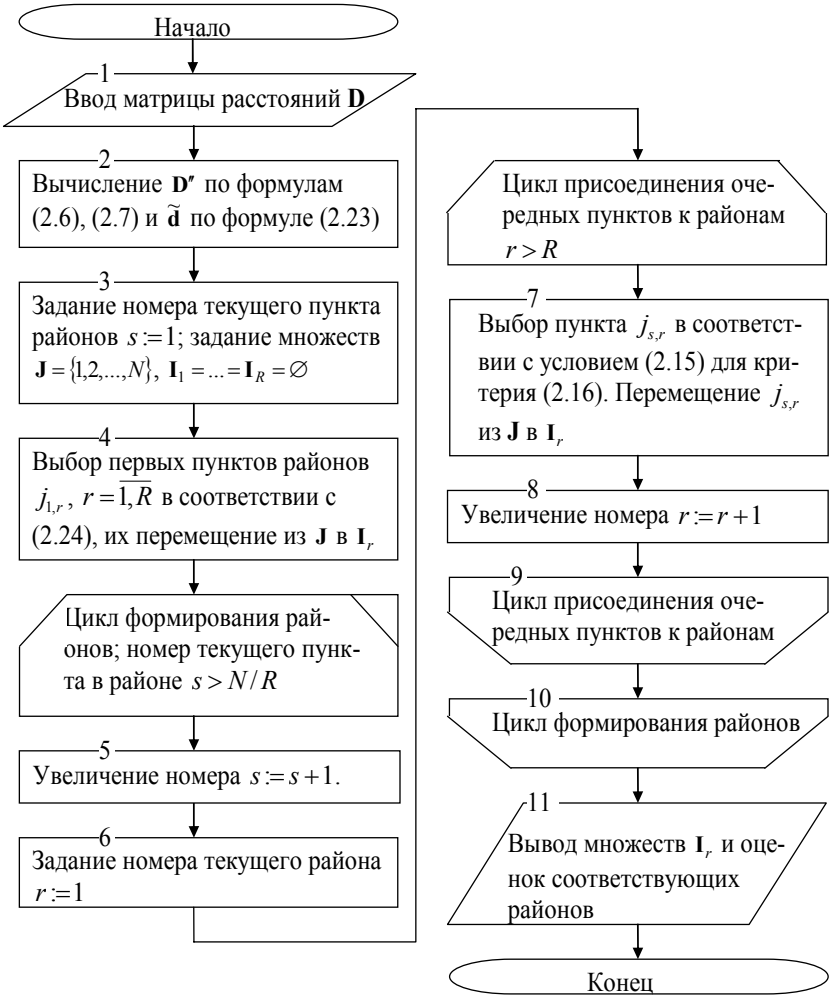


Рис. 2.4 − Схема алгоритма 2

2.3.4. Третий вариант алгоритма модифицированного метода «ветвей и границ»

Шаг 1. Вводится матрица расстояний $\mathbf{D}$.

Шаг 2. В соответствии с формулами (2.1), (2.2) формируется приведенная матрица $\mathbf{D}''$. Формируется вектор $\tilde{\mathbf{d}}$, элементы которого вычисляются по формуле (2.23).

Шаг 3. Устанавливается номер текущего района $r := 1$, множество пунктов, не присоединенных ни к одному из районов $\mathbf{J} = \{1,2,...,N\}$, множества входящих в районы пунктов $\mathbf{I}_1 = \mathbf{I}_2 = ... = \mathbf{I}_R = \varnothing$.

Шаг 4. Количество элементов в текущем районе приравнивается к единице, $s := 1$.

Шаг 5. Выбирается первый элемент $j_{1,r}$ района *r* в соответствии с условием (2.24).

Элемент $j_{1,r}$ перемещается из множества $\mathbf{J}$ в множество элементов района $\mathbf{I}_r$.

Шаг 6. Номер очередного элемента района увеличивается на единицу, $s := s + 1$.

Шаг 7. Для всех пунктов $j \in \mathbf{J}$ вычисляется критерий F_j

$$F_j = \sum_{i \in \mathbf{I}} d''_{ij} . \qquad (2.25)$$

Производится выбор пункта $j_{s,r}$, для которого выполняется условие

$$F_{j_{s,r}} = \min_{j \in \mathbf{J}} F_j . \qquad (2.26)$$

Выбранный пункт перемещается из множества $\mathbf{J}$ в множество $\mathbf{I}_r$.

Шаг 8. Если количество элементов в районе $s < \frac{N}{R}$, то осуществляется переход к шагу 6.

Шаг 9. Номер района увеличивается на 1, $r := r + 1$.

Шаг 10. Если $r \le R$, то производится переход к шагу 5.

Шаг 11. На экран выводятся списки пунктов доставки, которые рекомендуется отнести к каждому из районов, длины маршрутов коммивояжера для каждого полученного района и их суммарная длина.

Блок-схема алгоритма 3 приведена на рис. 2.5.

2.3.5. Четвёртый вариант алгоритма модифицированного метода «ветвей и границ»

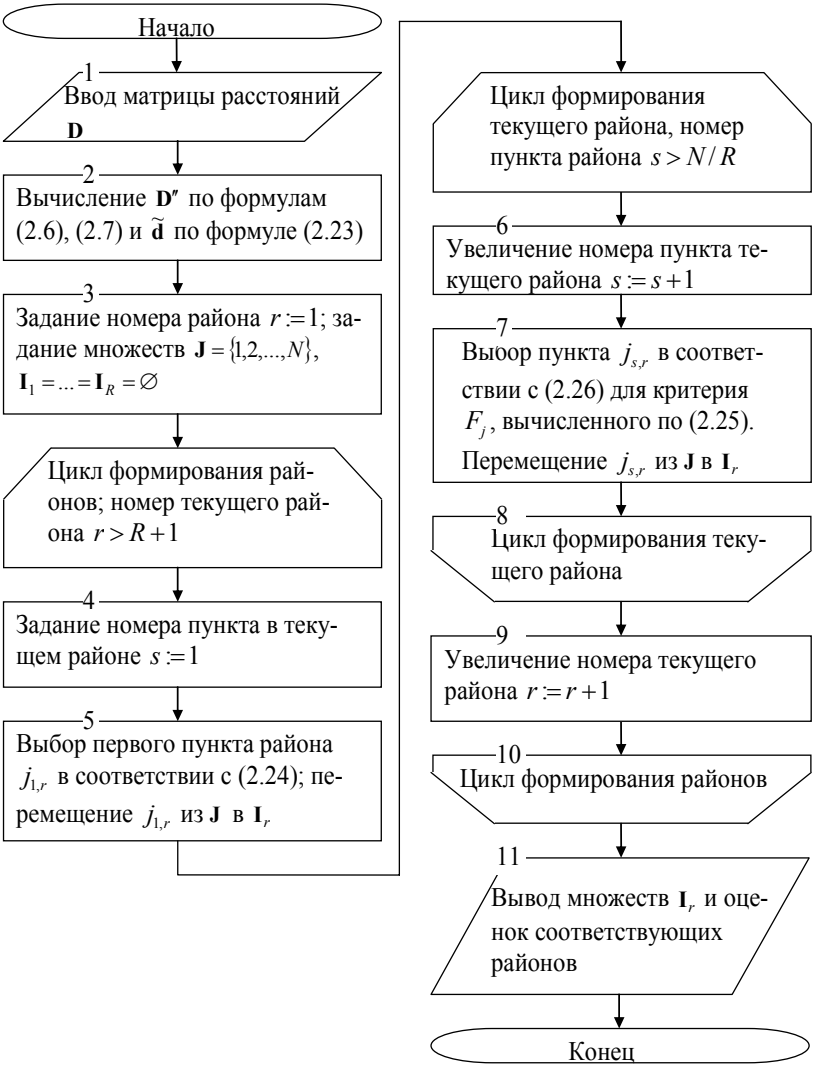
Шаг 1. Вводится матрица расстояний $\mathbf{D}$.

Шаг 2. В соответствии с формулами (2.6), (2.7) формируется приведенная матрица $\mathbf{D}''$. Формируется вектор $\tilde{\mathbf{d}}$, элементы которого вычисляются по формуле (2.23).

Шаг 3. Устанавливается номер текущего района $r := 1$, множество пунктов, не присоединенных ни к одному из районов $\mathbf{J} = \{1,2,...,N\}$, множества входящих в районы пунктов $\mathbf{I}_1 = \mathbf{I}_2 = ... = \mathbf{I}_R = \varnothing$.

Шаг 4. Количество элементов в текущем районе устанавливается равным единице $s := 1$.

Шаг 5. Выбирается первый элемент $j_{1,r}$ района *r* в соответствии с условием (2.24). Элемент $j_{1,r}$ перемещается из множества $\mathbf{J}$ во множество элементов района $\mathbf{I}_r$.



Р

Рис. 2.5 − Схема алгоритма 3

Шаг 6. Номер очередного элемента района увеличивается на единицу, $s := s + 1$.

Шаг 7. Если номер текущего пункта $s \geq \frac{N}{2R}$, то осуществляется переход к шагу 9.

Шаг 8. Исходя из условия (2.15) для критерия E_j, вычисляемого по формуле (2.16), выбирается элемент $j_{s,r}$. Осуществляется переход к шагу 10.

Шаг 9. Исходя из условия (2.19) для критерия E'_j, вычисляемого по формуле (2.17), выбирается элемент $j_{s,r}$.

Шаг 10. Выбранный пункт перемещается из множества $\mathbf{J}$ в множество $\mathbf{I}_r$.

Шаг 11. Если количество элементов в районе $s < \frac{N}{R}$, то мы переходим к шагу 6.

Шаг 12. Номер района увеличивается на единицу, $r := r + 1$.

Шаг 13. Если $r \leq R$, то осуществляется возврат к шагу 4.

Шаг 14. На экран выводятся списки пунктов доставки, которые рекомендуется отнести к каждому из районов, длины маршрутов коммивояжера для каждого полученного района и их суммарная длина.

Блок-схема алгоритма 4 приведена на рис. 2.6.

2.3.6. Пятый вариант алгоритма модифицированного метода «ветвей и границ»

Шаг 1. Водится матрица расстояний $\mathbf{D}$.

Шаг 2. В соответствии с формулами (2.6), (2.7) формируется приведенная матрица $\mathbf{D}''$. Формируется вектор $\tilde{\mathbf{d}}$, элементы которого вычисляются по формуле (2.23).

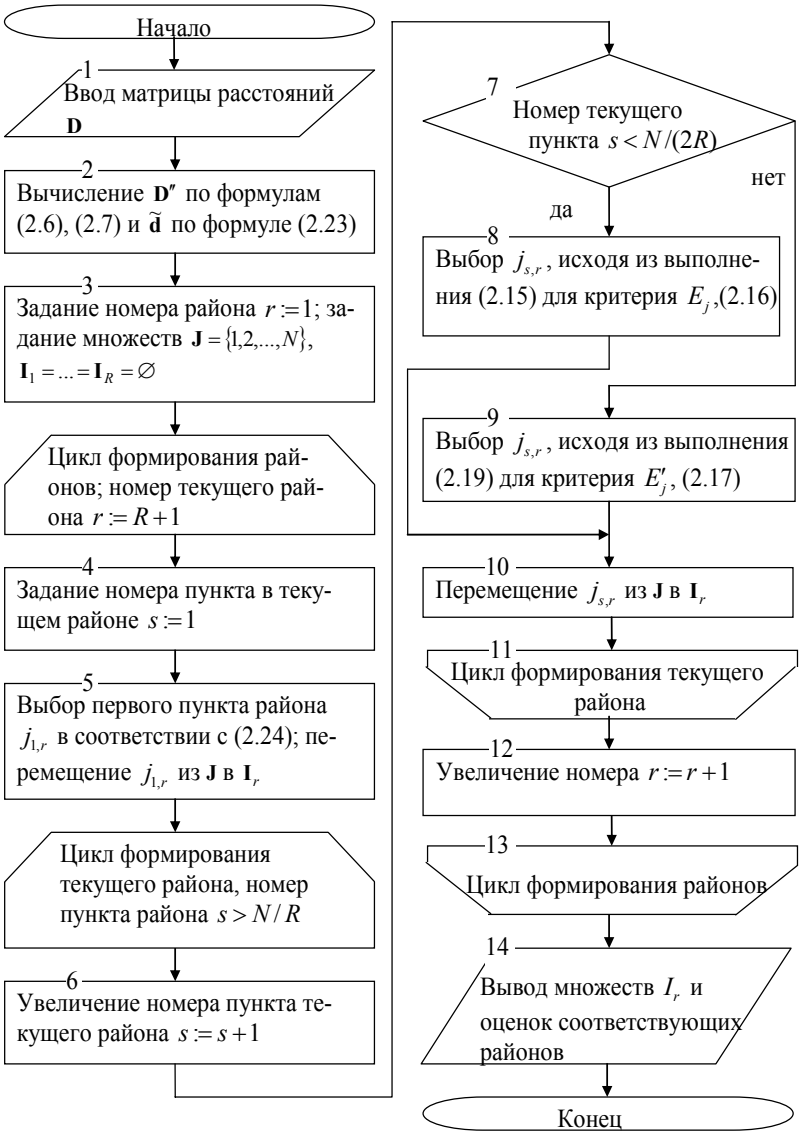


Рис. 2.6 − Схема алгоритма 4

Шаг 3. Устанавливается номер текущего района $r := 1$, множество пунктов, не присоединенных ни к одному из районов $\mathbf{J} = \{1,2,...,N\}$, множества входящих в районы пунктов $\mathbf{I}_1 = \mathbf{I}_2 = ... = \mathbf{I}_R = \varnothing$.

Шаг 4. Задается номер пункта в текущем районе $s := 1$.

Шаг 5. Выбирается первый элемент $j_{1,r}$ района *r* в соответствии с условием (2.19). Элемент $j_{1,r}$ перемещается из множества $\mathbf{J}$ в множество элементов района $\mathbf{I}_r$.

Шаг 6. Номер очередного элемента района увеличивается на единицу, $s := s + 1$.

Шаг 7. Если номер текущего пункта $s < \dfrac{N}{3R}$, то осуществляется переход к следующему шагу. Если $\dfrac{N}{3R} \le s < \dfrac{2N}{3R}$, то осуществляется переход к шагу 9. Если $s \ge \dfrac{2N}{3R}$, то осуществляется переход к шагу 10.

Шаг 8. Исходя из условия (2.10) для критерия E_j, вычисляемого по формуле (2.11), выбирается элемент $j_{s,r}$. Осуществляется переход к шагу 11.

Шаг 9. Исходя из условия (2.14) для критерия E'_j, вычисляемого по формуле (2.12), выбирается элемент $j_{s,r}$. Осуществляется переход к шагу 11.

Шаг 10. Исходя из условия (2.14) для критерия E'_j, вычисляемого по формуле (2.13), выбирается элемент $j_{s,r}$.

Шаг 11. Выбранный пункт $j_{s,r}$ перемещается из множества $\mathbf{J}$ в множество $\mathbf{I}_r$.

Шаг 12. Если количество элементов в районе $s < \dfrac{N}{R}$, то мы переходим к шагу 6.

Шаг 13. Номер района увеличивается на единицу: $r := r + 1$.

Шаг 14. Если номер района $r < R + 1$, то осуществляется возврат к шагу 4.

Шаг 15. На экран выводятся списки пунктов доставки, которые рекомендуется отнести к каждому из районов, длины маршрутов коммивояжера для каждого полученного района и их суммарная длина.

Блок-схема алгоритма 5 приведена на рис. 2.7.

2.3.7. Шестой вариант алгоритма модифицированного метода «ветвей и границ»

Шаг 1. Водится матрица расстояний $\mathbf{D}$.

Шаг 2. В соответствии с формулами (2.6), (2.7) формируется приведенная матрица $\mathbf{D}''$. По формуле (2.8) вычисляются оценки $L_{ср.}(j)$ для всех пунктов, $j = \overline{1, N}$.

Шаг 3. Устанавливается номер текущего района $r := 1$, множество пунктов, не присоединенных ни к одному из районов $\mathbf{J} = \{1, 2, ..., N\}$, множества входящих в районы пунктов $\mathbf{I}_1 = \mathbf{I}_2 = ... = \mathbf{I}_R = \varnothing$.

Шаг 4. Задается номер элемента в текущем районе $s := 1$.

Шаг 5. Если номер текущего района $r = 1$, то осуществляется переход к следующему шагу. В противном случае – к шагу 7.

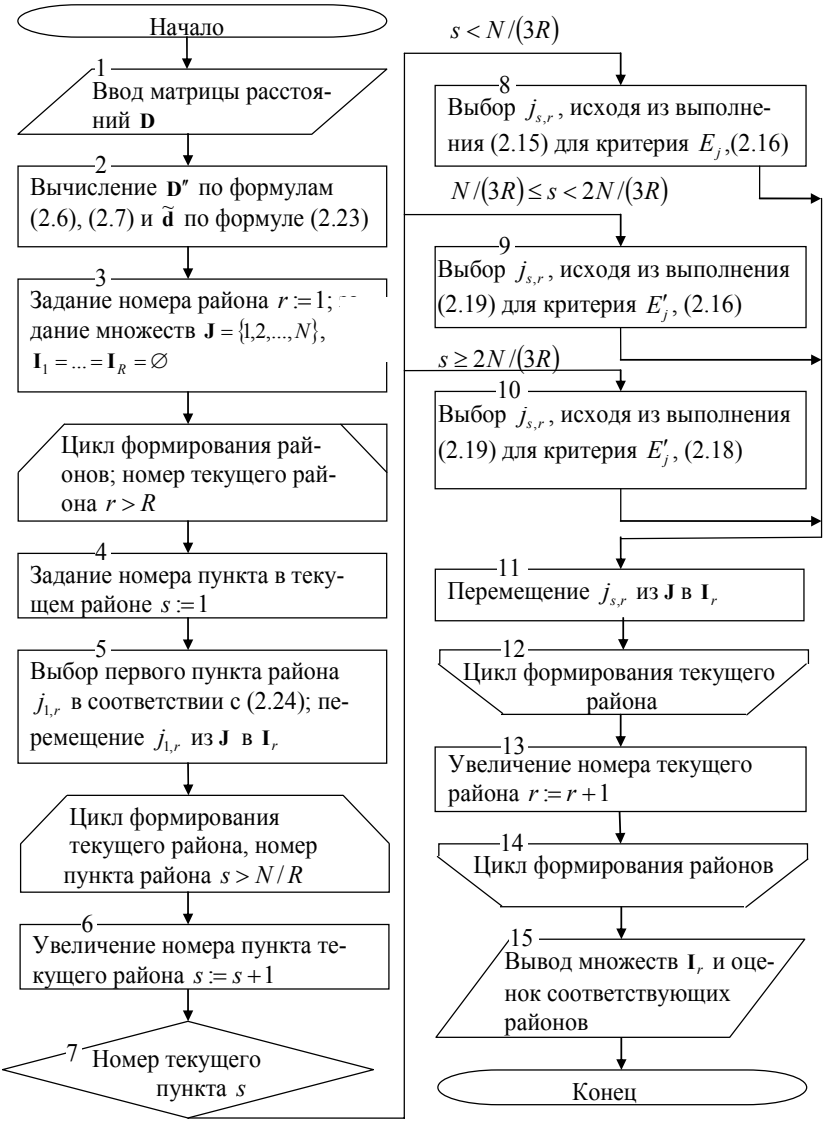


Рис. 2.7 − Схема алгоритма 5

Шаг 6. Выбирается первый элемент $j_{1,r}$ района *r* в соответствии с условием (2.9). Осуществляется переход к шагу 8.

Шаг 7. Выбирается первый элемент $j_{1,r}$ района *r* в соответствии с условием (2.13).

Шаг 8. Элемент $j_{1,r}$ перемещается из множества $\mathbf{J}$ в множество элементов района $\mathbf{I}_r$.

Шаг 9. Номер очередного элемента района увеличивается на единицу: $s := s + 1$.

Шаг 10. Исходя из условия (2.15) для критерия E_j, вычисляемого по формуле (2.16), выбирается элемент $j_{s,r}$.

Шаг 11. Выбранный пункт $j_{s,r}$ перемещается из множества $\mathbf{J}$ в множество $\mathbf{I}_r$.

Шаг 12. Если количество элементов в районе $s < \frac{N}{2R}$, то мы переходим к шагу 9.

Шаг 13. Номер района увеличивается на единицу, $r := r + 1$.

Шаг 14. Если номер района $r \le R$, осуществляется возврат к шагу 4.

Шаг 15. Выполняется присвоение номера района $r := 1$.

Шаг 16. Если количество пунктов в текущем районе не превышает $\left[\frac{N}{R}\right] + Np$, то происходит переход к следующему шагу. В противном случае осуществляется переход к шагу 21. Число *p* – это число процентов, на которое количество пунктов в текущем районе не должно превышать число $\left[\frac{N}{R}\right]$.

Шаг 17. Исходя из условия (2.19) для критерия E'_j, вычисляемого по формуле (2.17), выбирается элемент $j_{s+1,r}$.

Шаг 18. Для выбранного элемента $j_{s,r}$ выбирается ближайший район r_k исходя из условия (2.20) для критерия $E''_{r_t, j_{s+1,r}}$, вычисляемого по формуле (2.21).

Шаг 19. Если $r_k = r$, то осуществляется переход к следующему шагу. В противном случае – к шагу 21.

Шаг 20. Количество пунктов в рассматриваемом районе *r* увеличивается на 1, $s := s + 1$. Выбранный пункт перемещается из множества $\mathbf{J}$ в множество $\mathbf{I}_r$.

Шаг 21. Если $r < R$, то происходит переход к следующему шагу. В противном случае – к шагу 23.

Шаг 22. Номер района увеличивается на 1, $r := r + 1$. Осуществляется переход к шагу 24.

Шаг 23. Выполняется присвоение $r := 1$.

Шаг 24. В случае, если множество $\mathbf{J}$ не пустое, $\mathbf{J} \neq \varnothing$, то осуществляется возврат к шагу 16.

Шаг 25. На экран выводятся списки пунктов доставки, которые рекомендуется отнести к каждому из районов, длины маршрутов коммивояжера для каждого полученного района и их суммарная длина.

Блок-схема алгоритма 6 приведена на рис. 2.8 – 2.9

2.4. Анализ вариантов алгоритма модифицированного метода «ветвей и границ»

В таблице 2.1 приведены результаты тестирования рассмотренных вариантов алгоритма модифицированного метода «ветвей и границ» применительно к задаче разбиения на районы транспортных сетей малой размерности.

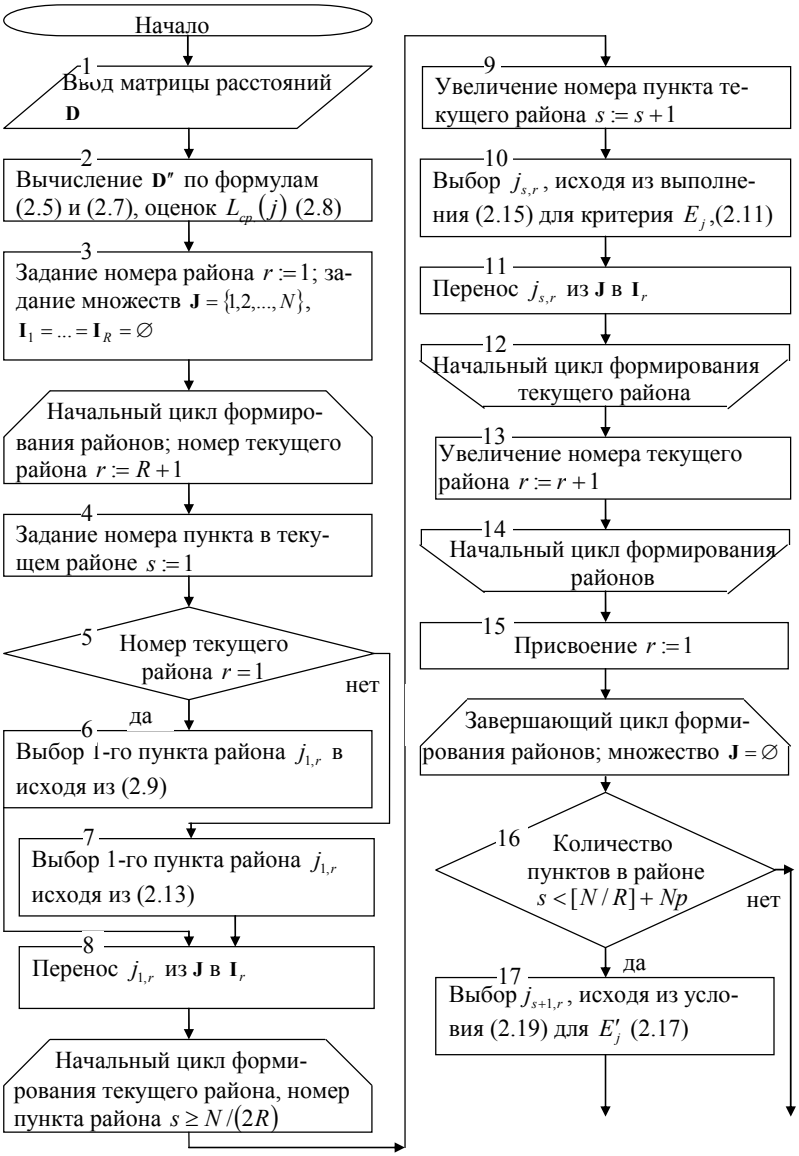


Рис. 2.8 – Начало схемы алгоритма 6

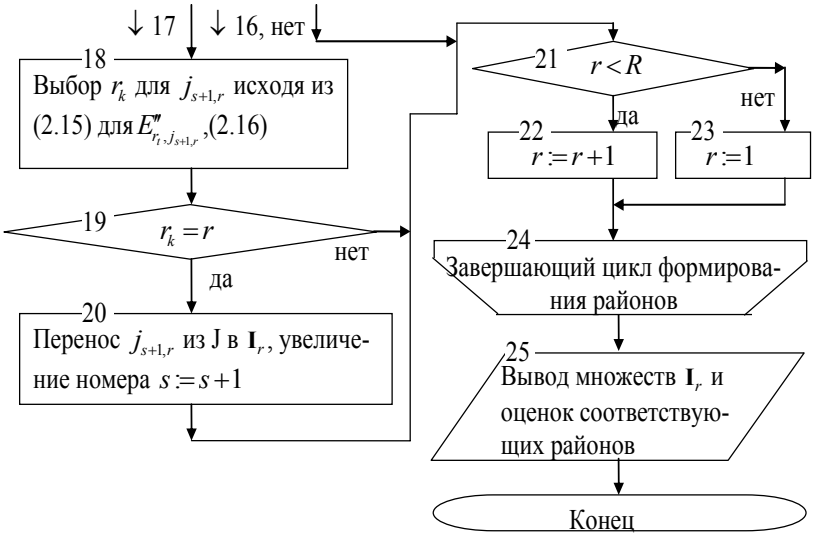


Рис. 2.9 – Окончание схемы алгоритма 6

Из анализа результатов тестирования следует, что наименьшее в среднем отклонение от оптимального результата дает алгоритм 4. Но при этом имеет место очевидная тенденция к увеличению отклонения от оптимального результата с увеличением количества пунктов в транспортной сети. У алгоритма 3 (второго по малости среднего значения) эта тенденция выражена еще более ярко. У алгоритма 6 (следующего по малости среднего значения) отклонение от оптимального значения возрастает меньше всего. Так как наиболее актуальным является решение задач именно больших размерностей, алгоритм 6 был выбран как наиболее подходящий для решения поставленной задачи при формировании окончательного варианта адаптации метода «ветвей и границ».

Таблица 2.1 – Результаты тестирования альтернативных алгоритмов модифицированного метода «ветвей и границ» на транспортных сетях малой размерности

Алгоритм разбиения	Отклонение от оптимального результата, %									
	Разбиение на 2 района						Разбиение на 3 района			Среднее значение
Общее кол-во пунктов завоза	8	10	12	14	16	18	9	12	15	
Алгоритм 1	1,2	3,6	10,5	6,9	10	0	5,7	3,7	0,6	4,7
Алгоритм 2	14,8	13	23,5	14,4	26,3	17,7	11,8	24,5	9,7	17,3
Алгоритм 3	0,7	0	5	4,5	5,9	10	1,8	0,4	0,6	3,20
Алгоритм 4	1,6	1,6	5	2,3	6,5	5,1	0	0	0	2,4
Алгоритм 5	0,7	1,6	5	8,4	8,2	8,2	8,2	1,1	0	4,6

РАЗДЕЛ 3

РЕШЕНИЕ ЗАДАЧИ КОММИВОЯЖЕРА БОЛЬШОЙ РАЗМЕРНОСТИ НА ОСНОВЕ ПРИНЦИПОВ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ

В настоящее время для доставки грузов широко применяется автомобильный транспорт. В связи с этим достаточно актуальной является задача организации доставки мелкопартионных грузов. Данная задача обычно сводится к задаче коммивояжера или задаче развозки.

При решении задачи коммивояжера обычно стремятся достигнуть минимальной длины маршрута, то есть минимизировать длину пробега автомобиля. Однако, используя те же методы, можно легко перейти к минимизации затрат времени, если учитывать не только длины путей, связывающих пункты отправления и доставки, но и скорость, с которой можно двигаться по соответствующим участкам дорог, и время, затрачиваемое на остановки в каждом пункте.

Задачу коммивояжера можно также рассматривать как вспомогательную при решении задач, конечным результатом которых оптимальный маршрут коммивояжера не является. Во втором разделе данной работы предложен алгоритм, при реализации и оценке качества которого необходимо выполнять построение маршрута коммивояжера достаточно много раз.

Важными критериями работы алгоритма для решения задачи коммивояжера являются как точность получаемых результатов, так и скорость их получения.

Существует достаточно много методов, обеспечивающих более или менее точное решение задачи коммивояжера. При их анализе затруднение вызывает тот факт, что последние подробные иссле-

дования по этой теме проводились в 80-е годы. В связи с этим трудно реально оценивать быстродействие всех точных и эвристических методов, описанных в литературе [5]. В то же время возникает вопрос о точности эвристических методов, предлагаемых в литературе, о влиянии размерности задачи на получаемую точность.

В ходе выполнения работы было протестировано быстродействие методов «ветвей и границ» и динамического программирования, которые по общему признанию позволяют получать точные результаты. Проведенное тестирование показало, что метод динамического программирования позволяет решать задачи более высокой размерности по сравнению с методом «ветвей и границ». Но при проверке оказалось, что он не является точным. «Минимальные» маршруты, полученные с его помощью, могут быть хуже маршрутов, полученных посредством метода «ветвей и границ».

3.1. Анализ существующих методов маршрутизации мелкопартионных перевозок

Как было отмечено выше, решение задачи маршрутизации мелкопартионных перевозок может быть сведено к решению задачи коммивояжера или к решению задачи развозки. Задача коммивояжера предполагает прохождение всех пунктов доставки в одном маршруте. В задаче развозки строится несколько маршрутов.

Методы решения указанных задач делятся на две основные группы. Первая из них – точные методы, которые гарантируют объективно оптимальные решения. Вторая – приближенные методы, которые позволяют приблизиться к оптимальному решению с заданной точностью приближения.

Одним из широко известных точных методов решения задачи коммивояжера является метод динамического программирования [15]. Данный метод предназначен для решения широкого круга задач, которые могут быть разбиты на перекрывающие друг друга подзадачи, причем оптимальное решение этих подзадач может быть использовано для решения исходной задачи. Вначале осуществляется решение подзадач, а далее на основании полученных решений находится решение исходной задачи.

Применительно к задаче коммивояжера подзадачей является начальная часть маршрута, включающая в себя некоторое количество пунктов меньше общего количества. Решение достигается за n итераций, где n – общее количество пунктов завоза.

Вначале находятся кратчайшие маршруты, проходящие через 1 пункт, потом, на их основании – проходящие через 2 пункта, и так далее. На последней итерации находятся маршруты, проходящие через все пункты. Итеративный процесс завершается оптимальным замыканием маршрута.

При поиске маршрута из пункта i в пункт 0, включающего k транзитных пунктов, $k=\overline{2,n}$, маршрут, содержащий $k-1$ транзитный пункт, выбирается из условия

$$f(i;j_1,j_2,...,j_k)=$$

$$=\min_{1\le m\le k}\left\{d_{ij_m}+f(j_m;j_1,j_2,...,j_{m-1},j_{m+1},...,j_k)\right\}. \quad (3.1)$$

где d_{ij_m} – длина звена транспортной сети между пунктами i и j_m, $f(i;j_1,j_2,...,j_k)$ – оценка длины маршрута из i в 0, проходящего через пункты $j_1,j_2,...,j_k$.

При выборе маршрута с $(k-1)$-м транзитным пунктом не рассматриваются маршруты, содержащие пункт i, $i\notin\{j_1,j_2,...,j_k\}$.

В случае, если на некотором шаге маршруты, соответствующие оценкам $f(j_m;j_1,j_2,...,j_{m-1},j_{m+1},...,j_k)$, $j_m=\overline{1,n}$, содержат пункт i, то есть не могут быть использованы для построения, используются маршруты, содержащие $k-2$ транзитных пункта, $k-3$ пункта и так далее.

Время работы данного алгоритма определяется эквипотенциальной величиной $O\left(n^2 2^n\right)$. В связи с особенностями алгоритма

является необходимым сохранение всех промежуточных вычислений. Это приводит к тому, что при реализации алгоритма требуемый объем памяти ЭВМ также возрастает в экспоненциальной зависимости от размерности задачи [16].

В [17] была предложена модификация метода динамического программирования, упрощающая поиск оптимального решения при помощи коррекции функций состояния. Коррекция позволяет заранее указать варианты, превышающие верхнюю границу, следовательно, заведомо неоптимальные, и исключить их из рассмотрения.

Для выбраковки элементов матрицы предлагается использовать отношение

$$p_{ij} = \frac{d_{i,j}}{\sum_{k=1}^{n} d_{i,k}}; \quad i = 1,2,...,n, \tag{3.2}$$

и оставлять в матрице заранее обусловленное количество элементов, имеющих наименьшее значение p_{ij}. Остальные элементы матрицы принимаются равными бесконечной величине. При этом отклонения от оптимальных планов составляют 1 – 1,5 %. Такая модификация позволяет находить квазиоптимальные решения для задач размерностью в 30 – 40 пунктов [17].

Среди точных методов наиболее широкое распространение получил метод «ветвей и границ» [5,18]. Метод «ветвей и границ» основан на разбиении множества маршрутов объезда всех пунктов транспортной сети на подмножества и постепенном отсеивании неперспективных подмножеств. Процесс продолжается до получения подмножества, содержащего единственный маршрут минимальной длины. Перспективными считаются подмножества, имеющие минимальную оценку, то есть совпадающей с точной нижней границей длины маршрута коммивояжера.

Графически разбиение представляется в виде дерева, вершинами которого являются полученные подмножества. Для вычисления оценок подмножеств данные о транспортной сети, содержащей

n пунктов, представляются в виде матрицы $\mathbf{D} = \{d_{ij}\}_{n \times n}$, в которую занесены все расстояния между пунктами транспортной сети.

Точная нижняя граница маршрута коммивояжера для всего множества маршрутов L равна сумме минимальных элементов каждой из строк матрицы $\mathbf{D}$ и минимальных элементов столбцов матрицы $\mathbf{D}$ после редуцирования по строкам:

$$L = \sum_{i=1}^{n} \min_{j \in \{1,\ldots n\}} d_{ij} + \sum_{j=1}^{n} \min_{i \in \{1,\ldots n\}} \left(d_{ij} - \min_{j \in \{1,\ldots n\}} d_{ij} \right). \qquad (3.3)$$

Редуцирование – это уменьшение всех элементов строки на величину ее наименьшего элемента. Редуцирование производится на каждом шаге для строк и столбцов получаемых матриц. Точная нижняя граница подмножеств оценивается аналогичным образом, но при этом используются соответствующие им матрицы.

При разбиении исходного множества и всех промежуточных подмножеств каждый раз получают два непересекающихся подмножества. Одно из подмножеств является множеством маршрутов, содержащих некоторое звено (i, j) транспортной сети, второе – множеством маршрутов, не содержащих этого звена. Точная нижняя граница длины маршрута коммивояжера для первого из подмножеств равна точной нижней границе L исходного множества, для второго из подмножеств – сумме $L + f_{(i,j)}$, где $f_{(i,j)}$ – оценка, вычисленная для звена (i, j) по формуле

$$f_{(i,j)} = \min_{q=1,n} \{d_{iq}\} + \min_{p=1,n} \{d_{pj}\}. \qquad (3.4)$$

Здесь числа d_{iq} и d_{pj} – это элементы матрицы расстояний после приведения.

Для разбиения на подмножества выбирается звено (i, j) с наибольшей оценкой $f_{(i,j)}$:

$$f_{(i,j)} = \max_{p,q}\left\{f_{(p,q)}\right\}. \qquad (3.5)$$

Из рассмотрения исключаются звенья, которые могут вызвать преждевременное зацикливание маршрута коммивояжера.

В случае, если на каком-либо этапе оказывается, что нижняя граница какого-либо из подмножеств, оставленных не разделенными, меньше, чем нижние границы вновь полученных подмножеств, полученный результат сохраняется и ветвление дерева продолжается из вершины, которой соответствует наименьшая нижняя граница.

Метод применим при решении задачи для 40 – 50 пунктов [5].

Метод «ветвей и границ» может быть также адаптирован для решения задачи развозки [5]. Для этого создается m копий пункта отправления груза, m соответствует требуемому количеству маршрутов. После решения задачи коммивояжера для такой расширенной транспортной сети копии совмещаются в один пункт. В результате получается необходимое количество маршрутов оптимальной длины. При необходимости в процессе построения маршрутов на каждом шаге анализируется общий объем заказа на груз в пунктах, включенных в маршрут. Маршруты, в которых объем заказа превышает грузоподъемность используемого транспортного средства, блокируются.

Иногда в целях экономии времени, затрачиваемого на решение, от поиска оптимума отказываются.

В работе [10] автор предлагает следующие пути для уменьшения затрат времени и необходимого объема используемой памяти ЭВМ:

- отказ от поиска точного решения;
- замена создания копий пункта отправления груза на разрешение многократного появления данного пункта в маршруте;

- блокирование в приведенной матрице максимальных элементов, появление которых в маршруте маловероятно;
- динамическое агрегирование пунктов доставки.

Еще один точный метод решения задачи коммивояжера был предложен А.В.Крушевским в [19]. Суть метода заключается в получении такого порядка пунктов назначения, который соответствует порядку их следования в маршруте минимальной длины. Для получения этого порядка применяется итеративный процесс. На каждом шаге по данным матрицы $\mathbf{D} = \{d_{ij}\}$, соответствующей текущему порядку пунктов, вычисляются вспомогательные величины

$$l(i,j) = d_{i-1j} + d_{ij+1} + d_{j-1i} + d_{ji+1} - d_{ii+1} - d_{i-1i} - d_{j-1j} - d_{jj+1},$$

$$j \neq i+1, \qquad (3.6)$$

$$l(i,j) = d_{i-1i} + d_{ii+2} + d_{i+1i} - d_{i-1i} - d_{ii+1} - d_{i+1i+2},$$

$$j = i+1, \quad i,j = \overline{2,n}. \qquad (3.7)$$

Величина $l(i,j)$ показывает, насколько увеличится или уменьшится длина маршрута при осуществлении перестановки местами элементов i и j.

После вычисления указанных оценок отбирается множество пар элементов $\{(i_1,j_1),(i_2,j_2),...,(i_s,j_s)\}$, которым соответствуют минимальные оценки

$$l(i_k,j_k) = \min_{\substack{i,j \\ i,j \notin \{i_1,j_1,..,i_{k-1},j_{k-1}\}}} \{l(i,j)\},\ k = \overline{1,s}. \qquad (3.8)$$

Для каждых двух пар элементов этого множества обязательным является выполнение условий

$$i_k \neq i_l,\ \ i_k \neq j_l,\ \ i_k \neq i_l+1,\ \ i_k \neq j_l+1,\quad k,l \in \{1,2,...,s\}.$$

Процесс продолжается до тех пор, пока не будет получен порядок элементов, соответствующий одному из полученных ранее.

Авторами [20] было исследовано сведение задачи коммивояжера к задаче линейного программирования.

Для составления математической модели задача вводятся булевы переменные

$$x_{ij} = \begin{cases} 1, & \text{если в маршруте присутствует переезд из} \\ & \text{пункта } i \text{ в пункт } j; \\ 0 & - \text{ в противном случае.} \end{cases}$$

Кроме того, вводятся целочисленные переменные u_i, u_j $\left(i, j = \overline{1,n}\right)$, принимающие произвольные значения.

Математическая модель задачи имеет вид:

$$\sum_{i=0}^{n}\sum_{j=0}^{n} d_{ij}x_{ij} \to \min_{x_{ij}\in\Omega}, \tag{3.9}$$

$$\Omega: \quad \sum_{\substack{i=0 \\ i\neq j}}^{n} x_{ij} = 1, \ j = \overline{1,n}; \tag{3.10}$$

$$\sum_{\substack{j=0 \\ j\neq i}}^{n} x_{ij} = 1, \ i = \overline{1,n}, \tag{3.11}$$

$$x_{ij} + x_{ji} \le 1, \ i, j = \overline{0,n}, \ i \neq j, \tag{3.12}$$

$$u_i - u_j + nx_{ij} \le n-1, \ i, j = \overline{1,n}, \ i \neq j. \tag{3.13}$$

За приемлемое время при помощи методов линейного программирования данную задачу можно решить только для количест-

ва пунктов завоза $n \leq 10$. Более того, при решении задачи на ЭВМ, возникает необходимость округления искомых величин. Это затрудняет решение и приводит к искажению результата [21].

К задаче целочисленного линейного программирования может быть сведена также и задача развозки [20]. Для получения решения задачи развозки система уравнений, используемая для решения задачи коммивояжера, дополняется ограничениями, запрещающими построение пересекающихся маршрутов. Это позволяет снизить количество необходимых вычислений и, следовательно, увеличить размерность задачи, которую можно решить за приемлемое время.

Так как применение точных методов возможно только для задач небольшой размерности (30-40 пунктов), более широкое применение получили приближенные методы.

Приближенные методы решения задачи развозки делятся на три группы: случайного поиска, локальной оптимизации и эвристические [5].

К методам случайного поиска относят микрорайонирование клиентов [22] и выделение типичных ситуаций [23]. Разработка данной группы методов была обусловлена наличием нестабильности спроса клиентов. В связи с нестабильностью спроса возникает неравномерность объемов завоза клиентам и периодичность завозов, тогда как для снабжающих организаций и автотранспорта желательно иметь стабильные планы работы.

Указанные методы предполагают разделение клиентов на группы в зависимости от регулярности поступающих от них заказов. Для выделенных типичных ситуаций формируется несколько возможных вариантов маршрута. После получения заказа диспетчером осуществляется выбор наиболее подходящего из планов или корректировка предварительно сформированных маршрутов.

Наиболее известным методом локальной оптимизации является метод инверсий [24]. Основная идея этого метода заключается в том, что имеющееся первоначальное решение разбивается на r фрагментов, которые затем снова объединяются в один маршрут, но при помощи звеньев, отсутствующих в исходном маршруте. При переборе всех обеспечивающих уменьшение длины маршрута вариантов замены r удаленных ребер можно получить маршрут,

который не может быть улучшен с помощью данной процедуры. Такой маршрут называется r-оптимальным.

В [25] приведено исследование 2-оптимальной процедуры. При выборе $r = 2$ для восстановления маршрута существуют только два звена и один вариант объединения маршрута. При этом один из фрагментов будет пройден в противоположном первоначальному направлении. Если из маршрута удаляются звенья $(i, i+1)$ и $(j, j+1)$ при $(i+1) < j$, то восстановить маршрут можно путем включения в него звеньев (i, j) и $(i+1, j+1)$. При этом сокращение длины маршрута от такой операции оценивается выражением

$$f_{i-j} = \sum_{k=i+1}^{j-1} d_{k,k+1} + d_{i,i+1} + d_{j,j+1} - \sum_{k=i+1}^{j-1} d_{k+1,k} - d_{i,j} - d_{i+1,j+1}. \quad (3.14)$$

Выбор значения r, равного 2, гарантирует наименьшие затраты времени для решения задачи, однако полученное решение превышает оптимальное на 4,5% [24].

Согласно [24], при выборе $r = 3$ или $r = 4$ полученное решение отличается от оптимального на 1,1%. Однако при этом значительно увеличивается объем вычислений. Если для 2-оптимальных планов на каждом шаге необходимо анализировать $n(n-3)$ вариантов замены звеньев, то для 3-оптимальных планов – уже $4n(n-4)(n-5)/3$ вариантов. По оценке [5] метод инверсий для 3-оптимальных планов требует в 4 раза более трудоемких вычислений, чем, например, «экономизирующий» метод Кларка-Райта.

Для сокращения количества рассматриваемых вариантов в работе [26] было предложено накладывать дополнительные ограничения при замене.

Пусть при реализации метода осуществляется замена ребер (i_k, j_k) на ребра (l_k, m_k), $k = \overline{1, r}$. Тогда перестановка выполняется только при условии выполнения условия

$$\sum_{k=1}^{r}\left(d_{i_k,j_k}-d_{l_k,m_k}\right)>0\,. \qquad (3.15)$$

В [27] указан способ, как улучшить полученный 3-оптимальный план при помощи перестановки пунктов внутри маршрутов и между маршрутами. При выполнении данных перестановок выгода оценивается следующим выражением

$$f_{i-j}=d_{i-1,j}+d_{i,i+1}+d_{j,j+1}-d_{i-1,i+1}-d_{i,j+1}-d_{j,i}\,, \qquad (3.16)$$

где i – порядковый номер на маршруте переставляемого пункта; j – порядковый номер на маршруте пункта, после которого вставляется i-й пункт.

Наиболее распространенными методами решения задачи развозки являются эвристические методы, которые можно разделить на три группы. К первой группе относятся методы, моделирующие действие опытного диспетчера – планировщика маршрутов; ко второй – реализующие эвристику (формализованное субъективное понятие о «лучшем» маршруте); к третьей – имеющие в своей основе точные методы решения и использующие их свойства и особенности для упрощения вычислений и/или укорачивающие продолжительность расчетов ценой отказа от гарантий найти точное решение.

Одним из методов первой группы является метелочный метод [28]. Метод использует графическое представление транспортной сети. Определяется расположение i-го пункта, $i=\overline{2,n}$, относительно начального, $i=1$, в полярных координатах, то есть находится длина радиус-вектора r_i и угол

$$An_i=arctg\frac{y_i-y_1}{x_i-x_1}\,, \qquad (3.17)$$

где (x_i,y_i) – декартовы координаты i-го пункта.

Начальный план формируется на основании ранжирования пунктов по величине угла An_i с учетом грузоподъемности доступных транспортных средств. Далее осуществляются перестановки пунктов между маршрутами с целью минимизации суммарной длины маршрута.

Метелочный метод достаточно прост в вычислительном аспекте, но качество полученных решений низкое. Поэтому он обычно используется в сочетании с методами локальной оптимизации, позволяя очень быстро формировать исходное решение [22, 24, 26]. Существует также метод наибольшего угла, сходный по свойствам и также использующий графическую интерпретацию задачи развозки [29].

Чаще всего для решения задач развозки применяется метод Кларка-Райта, или «экономизирующий» [30]. На основании данного метода начальная система маршрутов преобразуется таким образом, чтобы каждое отдельное преобразование давало наибольшее улучшение. В качестве показателя улучшения Кларком и Райтом принята экономия пробега. Для ее оценки вычисляется матрица экономии, элементы которой

$$s_{ij} = d_{0,i} + d_{0,j} - d_{i,j}, \tag{3.18}$$

где $d_{0,i}$ – расстояние меду поставщиком и пунктом i, $d_{0,j}$ – расстояние между поставщиком и пунктом j, $d_{i,j}$ – расстояние между пунктами i и j.

На каждом шаге производится объединение двух маршрутов. После этого из матрицы вычеркиваются те экономии, которые уже не могут быть реализованы. Процесс прекращается, когда все положительные экономии вычеркнуты.

В [10] приведены ещё несколько способов расчета экономии, предложенные Т.Гаскелем:

$$\lambda_{ij} = S_{ij}\left(d + \left|d_{0,i} - d_{0,j}\right| - d_{i,j}\right), \tag{3.19}$$

где d – среднее расстояние от поставщика до пунктов завоза;

$$\pi_{ij} = S_{ij} - d_{i,j}; \tag{3.20}$$

$$M_{ij} = S_{ij} - \Theta d_{i,j}, \tag{3.21}$$

где Θ – параметр структуризации маршрута, оптимальная величина которого определяется для каждого конкретного района перевозок путем сопоставления решений, полученных при разных значениях Θ.

Т. Гаскелем были также рассмотрены последовательный и параллельный способы составления маршрутов. Последовательный способ подразумевает, что построение нового маршрута начинается только тогда, когда исчерпаны возможности по улучшению предыдущего или когда этот маршрут уже нельзя наращивать. При параллельном способе все маршруты наращиваются одновременно.

Сравнение различных способов расчета экономии и способов построения маршрутов показало [31], что лучшие результаты дает параллельный способ при расчете экономии по формуле (3.18), хотя и эти решения могут быть улучшены методами локальной оптимизации [5]. В связи с простотой производимых вычислений этот метод работает достаточно экономично, что и обусловило широкое распространение его разнообразных версий [32, 33].

Наиболее экономичные системы маршрутов могут быть получены при помощи точных методов. Однако все они предъявляют высокие требования к памяти ЭВМ, в связи с чем неприменимы для решения задачи маршрутизации в крупных транспортных сетях. Точные методы применимы только для сетей, содержащих не более чем 40 – 50 пунктов доставки.

В связи с этим более широкое распространение получили приближенные методы. Методы, основанные на субъективной оценке «лучшего» маршрута или на попытке смоделировать действия опытного диспетчера, достаточно просты в вычислительном аспекте и не требуют слишком больших затрат памяти ЭВМ. Однако часто такие методы дают решение, сильно отклоняющееся от оптимального.

Более устойчивыми к изменению условий задачи являются приближенные методы, основанные на точных. В таких модификациях уменьшение затрат памяти ЭВМ осуществляется за счет отказа от абсолютного оптимума и уменьшения числа рассматриваемых вариантов маршрутов. Часто, несмотря на отсечение многих вариантов, такие методы приводят к получению точного решения. Но гарантий получения такого решения нет.

Таким образом, актуальным является разработка методов маршрутизации, сочетающие в себе высокую точность и возможность применения для транспортных сетей большой размерности. Наиболее перспективными являются исследования возможности адаптации точных методов к решению задач большой размерности.

3.2. Метод динамического программирования и анализ его эффективности

Предлагаемый в настоящей работе алгоритм основан на методе динамического программирования, изложенном в [15]. Он позволяет получать более точные результаты и решать задачи большей размерности по сравнению с классическим методом динамического программирования.

Опишем суть метода динамического программирования применительно к задаче коммивояжера. Содержательная постановка задачи следующая: *построить кольцевой маршрут минимальной длины, начинающийся и заканчивающийся в начальном пункте* 0 *и проходящий через* n *других пунктов ровно по одному разу.*

3.2.1. Формирование матрицы расстояний между пунктами транспортной сети

При решении задачи коммивояжера принято считать, что рассматривается транспортная сеть, в которой существует путь из каждого пункта в любой другой. На практике, если для некоторых двух пунктов такого пути нет, то строится соединяющий их маршрут минимальной длины, проходящий через некоторые транзитные пункты.

Для примера рассмотрим транспортную сеть, содержащую 8 пунктов с двусторонним движением на каждом участке дорог, соединяющих эти пункты (рис. 3.1).

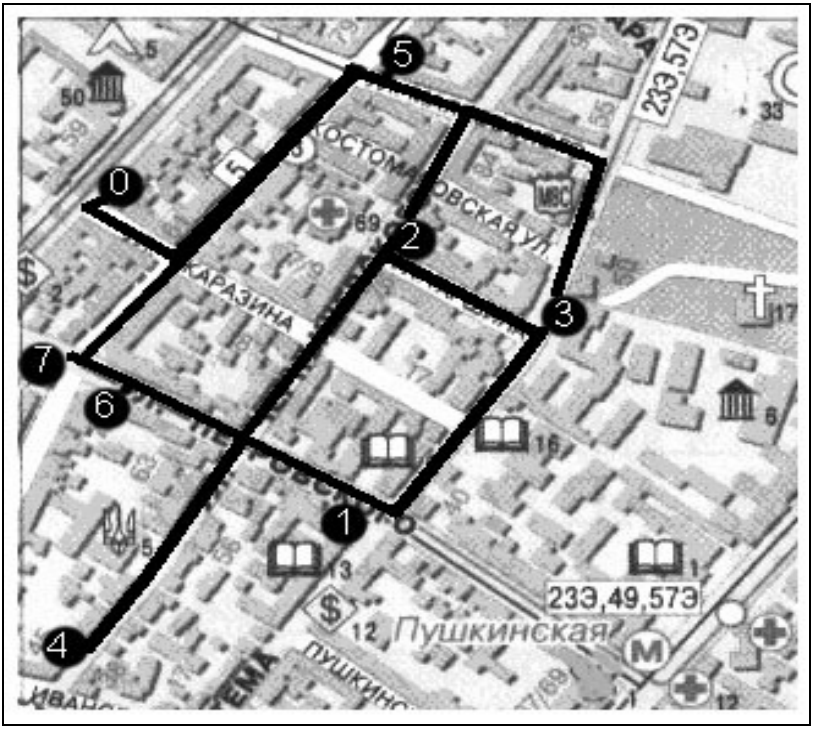


Рис. 3.1 – Схема транспортной сети, содержащей 8 пунктов

В транспортной сети на рис. 3.1 целесообразно задать расстояния между пунктами 0 и 5, 0 и 7, 1 и 2, 1 и 3, 1 и 4, 1 и 6, 2 и 3, 2 и 4, 2 и 5, 2 и 6, 3 и 5, 4 и 6, 5 и 7, 6 и 7, так как между ними существует маршрут, не проходящий через какие-либо другие пункты. Соответственно, между теми парами пунктов, которые отсутствуют в списке, длины путей заданы не будут. Расстояния между пунктами в данной транспортной сети однозначно представляются с помощью матрицы, соответствующей табл. 3.1.

Таблица 3.1 – Длины путей между пунктами транспортной сети без транзитных пунктов

	0	***1***	***2***	***3***	***4***	***5***	***6***	***7***
0						406		288
1			452	245	495		390	
2		452		205	611	350	445	
3		245	205			500		
4		495	611				460	
5	406		350	500				447
6		390	445		460			48
7	288					447	48	

В данном примере очевидно, что маршрут минимальной длины из пункта 0 в 6 проходит через пункт 7, его длина равна 288+48 = 336, оптимальный маршрут из пункта 0 в 1 (0, 7, 6, 1) имеет длину 288+48+390 = 726, оптимальный маршрут из пункта 0 в пункт 4 также проходит через пункты 7 и 6 и имеет длину 288+48+460 = 796. Уже не столь очевидно, но тоже достаточно просто можно определить длины оптимальных маршрутов из пункта 0 в 2 и из 0 в 3.

При выборе маршрута из пункта 0 в 2 рассмотрим варианты (0, 5, 2) и (0, 7, 6, 2). Длины этих маршрутов соответственно равны 756 и 781. Длина кратчайшего маршрута из 0 в 2 определится меньшим из этих чисел, то есть 756. Аналогичным образом можно установить, что длина оптимального маршрута из пункта 0 в пункт 3 равна 906, а сам маршрут соответствует последовательности пунктов (0, 5, 3).

Нетрудно получить маршруты для других пар пунктов, которые не связаны между собой непосредственно. В более сложных случаях используются специальные методы построения кратчайших маршрутов между пунктами, например, матричный.

В табл. 3.2 приведены все расстояния между пунктами данной сети, то есть длины путей между пунктами, связанными непосред-

ственно, и длины маршрутов между теми парами, для которых такая связь отсутствует.

Таблица 3.2 – Расстояния между пунктами транспортной сети

	0	***1***	***2***	***3***	***4***	***5***	***6***	***7***
0		726	756	906	796	406	336	288
1	726		450	245	495	745	390	438
2	756	450		205	611	350	445	493
3	906	245	205		740	500	635	683
4	796	495	611	740		955	460	508
5	406	745	350	500	955		495	447
6	336	390	445	635	460	495		48
7	288	438	493	683	508	447	48	

Как видно на примере пунктов 1 и 2, более выгодным с точки зрения длины оказывается не непосредственно связывающий эти пункты путь, а маршрут, проходящий через пункт 3.

3.2.2. Алгоритм метода динамического программирования применительно к задаче коммивояжера

Процесс формирования искомого маршрута осуществляется за n итераций, где n – количество пунктов в транспортной сети без учета начального (он же конечный) пункта в кольцевом маршруте коммивояжера, то есть без нулевого пункта. На каждой итерации формируется n маршрутов, $M_1, M_2, ..., M_n$, каждый из которых включает на 2 пункта больше, чем номер итерации. Маршрут $M_{k,i}$ на k-й итерации, $i = \overline{1,n}$, $k = \overline{1,n-1}$, может быть описан следующим образом. Он начинается в пункте i, проходит через k отличных от i пунктов и заканчивается в пункте 0. Маршрут формируется на основании маршрутов, полученных на предыдущей итерации. В формируемые маршруты каждый из пунктов должен входить не более одного раза.

На первой итерации для каждого значения i вычисляется функция, оценивающая длину пути из пункта i в конечный пункт 0 через пункт завоза $j,\ j \in \{1,...,n,\ j \neq i\}$:

$$f(i;j) = \min_{j \in \{1,...,n,\ j \neq i\}} \{d_{ij} + d_{j0}\}, \qquad (3.22)$$

где d_{ij} – расстояние между пунктами i и j.

На рис. 3.2, 3.3 приведены схемы, отображающие соответственно первую и вторую итерации метода динамического программирования для транспортной сети на рис. 3.1. Расстояния между пунктами в данной транспортной сети приведены в табл. 3.2.

Для пункта 1 транспортной сети на рис. 3.2 пунктирными линиями показаны все возможные маршруты, а жирной сплошной линией показан маршрут $M_{1,1} = (1,6,0)$, выбранный в соответствии с формулой (3.22):

$$f(1;6) =$$
$$= \min\left\{\underbrace{450+756}_{(1,2,0)}, \underbrace{245+906}_{(1,3,0)}, \underbrace{495+796}_{(1,4,0)}, \underbrace{745+406}_{(1,5,0)}, \underbrace{390+336}_{(1,6,0)}, \underbrace{438+288}_{(1,7,0)}\right\} =$$
$$= 726.$$

Минимальную длину имеют два последних рассматриваемых маршрута, $(1,6,0)$ и $(1,7,0)$, но в соответствии с методом динамического программирования должен быть выбран только один.

На рис. 3.2 показаны также маршруты, идущие из пунктов 2, 3, …, 7 в пункт 0 и проходящие через 1 транзитный пункт. В рамочке над каждым из пунктов указывается определяющая длину маршрута функция:

$$f(2;5) = 756,\quad f(3;5) = 906,\quad f(4;6) = 796,\quad f(5;7) = 735,$$
$$f(6;7) = 336,\quad f(7;6) = 384.$$

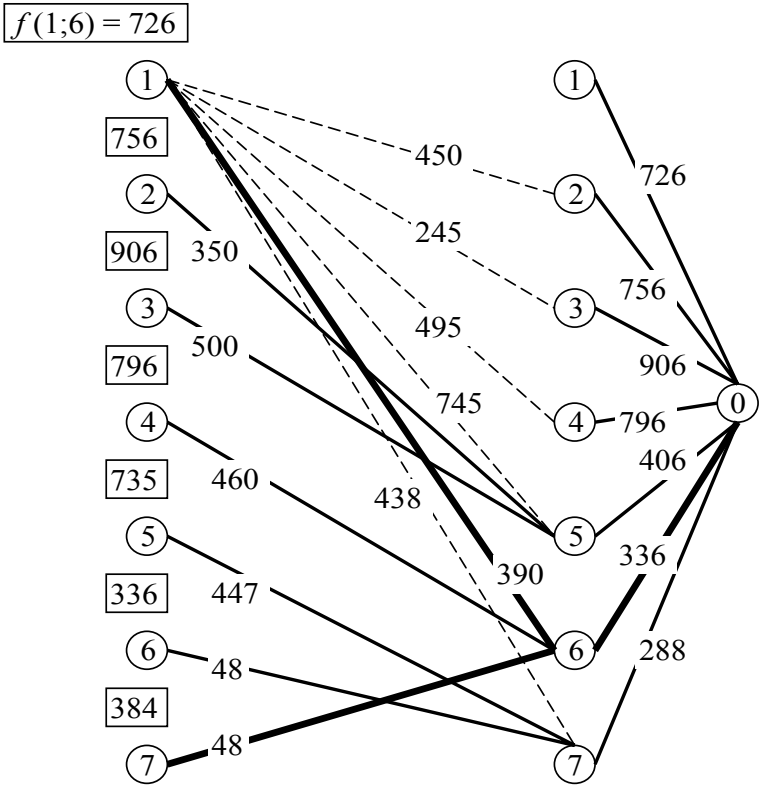


Рис. 3.2 – Схематическое изображение первой итераций метода динамического программирования для конкретной транспортной сети

Критерий оптимальности маршрута с произвольным количеством транзитных пунктов также базируется на функции, определяющей длину пути из пункта i в пункт 0:

$$f(i;\ j_1,\ j_2,...,j_k) = f(M_i) = \\ = \min_{1\le m\le k}\left\{d_{ij_m} + f(j_m;\ j_1,\ j_2,...,j_{m-1},\ j_{m+1},...,j_k)\right\}, \quad (3.23)$$

где $i = \overline{1,n}$, $\{j_1, j_2,..., j_k\} \subseteq \{1,2,...,i-1,i+1,...,n\}$.

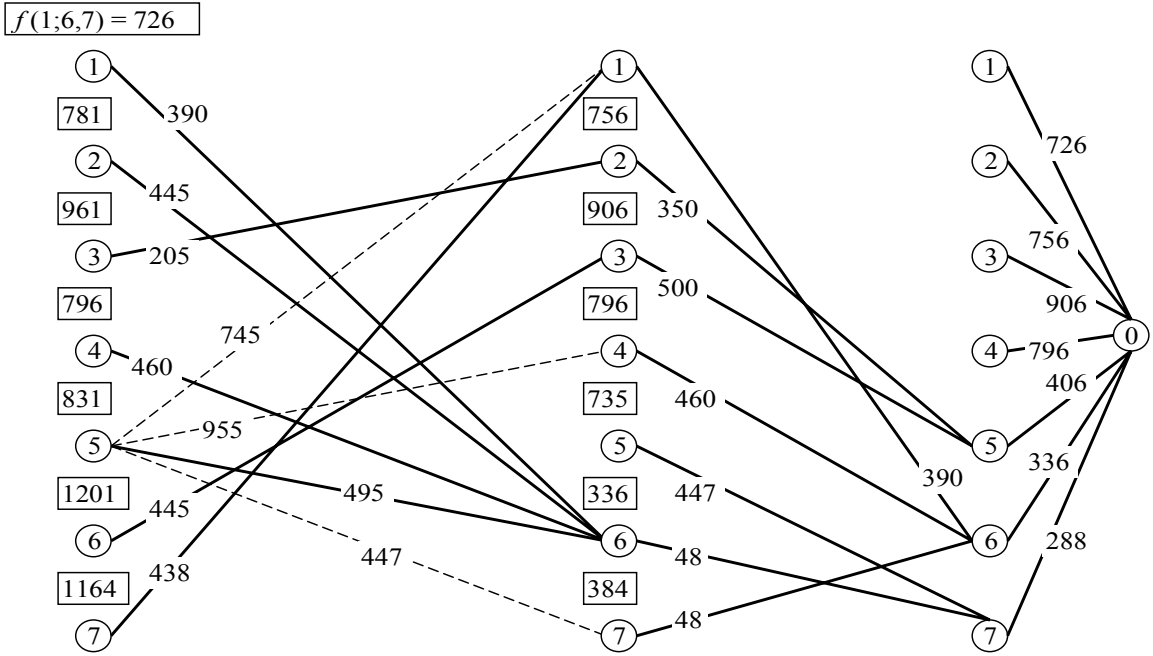


Рис. 3.3 – Схематическое изображение второй итерации метода динамического программирования для конкретной транспортной сети

Для примера снова рассмотрим транспортную сеть с рис. 3.1. На рис. 3.3 показана схема выбора в этой сети маршрута из пункта 5 в пункт 0 с двумя транзитными пунктами, то есть соответствующего второй итерации. Значение определяющей длину маршрута функции для пункта 5 на второй итерации

$$f(5;6,7) = \min\{745 + f(1;6), 955 + f(4;6), 495 + f(6;7), 447 + f(7;6)\} = 831.$$

Таким образом, маршрут из пункта 5 в пункт 0 на второй итерации определяется как $M_{2,5} = (5,6,7,0)$. Очевидно, что при выборе этого маршрута мы исключили из рассмотрения маршруты первой итерации $M_{1,2}$ и $M_{1,3}$, так как они включают в себя пункт 5.

Может возникнуть ситуация, когда на k-й итерации будет невозможно построить начинающийся в пункте i маршрут на основании результатов итерации $k-1$. Это происходит тогда, когда все маршруты $M_{k-1,1}, M_{k-1,2}, ..., M_{k-1,n}$ включают в себя пункт i: $M_{k-1,1}(i), M_{k-1,2}(i), \dots, M_{k-1,n}(i)$.

В качестве примера рассмотрим опять транспортную сеть, которая показана на рис. 3.1. При построении маршрутов в соответствии с методом динамического программирования на 4-й итерации имеем маршруты: $M_{4,1} = (1,7,6,2,5,0)$, $M_{4,2} = (2,3,1,6,7,0)$, $M_{4,3} = (3,2,1,6,7,0)$, $M_{4,4} = (4,1,3,2,5,0)$, $M_{4,5} = (5,3,1,6,7,0)$, $M_{4,6} = (6,1,3,2,5,0)$, $M_{4,7} = (7,1,3,2,5,0)$. Очевидно, что при переходе к итерации 5 на основании данных маршрутов невозможно будет построить маршрут, начинающийся в пункте 1, так как все полученные на 4-й итерации маршруты содержат данный пункт.

В этом случае осуществляется возврат к итерации $k-2$. На основании маршрутов $M_{k-2,1}, ..., M_{k-2,n}$ строятся альтернативные маршруты $M'_{k-1,1}, ..., M'_{k-1,n}$, которые не проходят через пункт i, и искомый маршрут $M_{k,i}$ формируется на основании альтернативных маршрутов.

Для транспортной сети в условиях примера (рис. 3.1) на 3-й итерации метода динамического программирования имеем несколько маршрутов, не проходящих через пункт 1: $M_{3,5}=(5,2,6,7,0)$, $M_{3,6}=(6,3,2,5,0)$, $M_{3,7}=(7,6,2,5,0)$, $f(M_{3,5})=1131$, $f(M_{3,6})=1596$, $f(M_{3,7})=1249$. Используя эти маршруты, можно построить $M'_{4,3}=(3,5,2,6,7,0)$, $M'_{4,4}=(4,7,6,2,5,0)$, $f(M'_{4,3})=1631$, $f(M'_{4,4})=1757$.

Таким образом, применяя результаты 3-й итерации, может быть построен маршрут $M_{5,1}=(1,3,5,2,6,7,0)$.

Если все маршруты $(k-2)$-й итерации тоже включают пункт i, то происходит возврат к итерации $k-3$, и так далее. Для этого на протяжении всей реализации метода необходимо сохранять результаты, полученные на всех шагах от первого до последнего.

На последней итерации с номером $k=n$ строится маршрут $M_{n,0}$, который начинается в пункте 0, проходит через пункты $1,2,...,n$ и заканчивается в пункте 0. Для этого среди маршрутов $M_{n-1,1},...,M_{n-1,n}$ выбирается тот, для которого реализуется условие

$$f(M_{n,0})=\min_{1\le m\le n}\{d_{0m}+f(M_{n-1,m})\}, \qquad (3.24)$$

или иначе

$$f(0;1,2,...,n)=\min_{1\le m\le n}\{d_{0m}+f(m;1,2,...,m-1,m+1,...,n)\}. \qquad (3.25)$$

В [15] утверждается, что метод динамического программирования является оптимальным. Однако при сравнении результатов, полученных при помощи данного метода, с результатами реализации метода «ветвей и границ» выяснилось, что «оптимальные» маршруты динамического программирования в среднем на 2,9 % длиннее полученных при помощи метода «ветвей и границ».

3.2.3. Тестирование метода динамического программирования применительно к решению задачи коммивояжера

Для тестирования методов решения задачи коммивояжера в среде *Delphi* 5 написаны специальные программы, реализующие метод «ветвей и границ» и метод динамического программирования.

В процессе тестирования найдены длины маршрутов коммивояжера для 40 различных графов с количеством вершин от 8 до 18. Вершины выбирались произвольным образом, расстояния между ними измерялись по прямой. Длины маршрутов, полученные при помощи метода динамического программирования, в ряде случаев оказывались больше длин маршрутов, полученных при помощи метода «ветвей и границ». В табл. 3.3 приводится среднее и максимальное отклонения длин этих маршрутов при различных размерностях транспортных сетей.

Таблица 3.3 – Отклонение длин маршрутов, полученных методом динамического программирования от полученных методом «ветвей и границ»

Относительный показатель	Количество пунктов в транспортной сети		
	8 — 10	12 — 14	15 — 18
Среднее отклонение, %	2,3	1,5	4,2
Максимальное откл., %	20,9	4,6	10

Более подробные данные о транспортных сетях, на которых производилось тестирование, и результатах самого тестирования приводятся в табл. 3.4.

Кроме того, что метод динамического программирования не гарантирует получения точных результатов, он имеет еще два существенных недостатка.

Во-первых, из-за необходимости сохранять всю промежуточную информацию от первой до последней итерации требуется большой объем динамической памяти ЭВМ.

Во-вторых, в случае возврата на предыдущие итерации существенно увеличивается время решения задачи.

Таблица 3.4 – Результаты тестирования метода динамического программирования

№ задачи	Расстояние между пунктами			Кол-во пунктов	Длина маршрута коммивояжера		Относительное отклонение
	наименьшее	наибольшее	среднее		оптимальная	полученная ДП	
1	131	1073	522,32	8	2563	2580	0,0066
2	133	837	436,18	8	2441	2950	0,2085
3	48	955	544,68	8	2497	2497	0
4	157	1273	681,25	8	2980	2980	0
5	229	1405	800,11	8	3888	3888	0
6	167	761	422,66	8	2213	2213	0
7	66	874	400,89	8	2195	2203	0,0036
8	69	1698	724,64	8	3873	3873	0
9	43	1081	518,43	8	2676	2676	0
10	78	677	357,25	8	1797	1797	0
11	154	1225	700,14	8	2899	2899	0
12	159	913	480,89	8	2372	2690	0,1341
13	212	1012	592,68	8	3389	3596	0,0611
14	11	88	48,68	8	244	246	0,0082
15	0	75	39,25	9	192	192	0
16	1	65	28,97	9	157	157	0
17	2	45	20,36	9	141	141	0
18	8	98	49,84	10	264	270	0,0227
19	15	165	68,07	10	454	479	0,0551
20	11	108	58,4	10	323	335	0,0372

Окончание табл. 3.4

№ задачи	Расстояние между пунктами			Кол-во пунктов	Длина маршрута коммивояжера		Относительное отклонение
	наименьшее	наибольшее	среднее		оптимальная	полученная ДП	
21	1	9	4,37	10	22	23	0,0455
22	1	28	13,87	10	97	98	0,0103
23	1	46	23,5	10	144	144	0
24	115	904	481,93	10	2619	2619	0
25	110	955	451,39	10	2490	2505	0,006
26	75	1080	490,22	10	2299	2299	0
27	5	57	29,12	12	231	238	0,0303
28	145	2128	866,5	12	4945	4945	0
29	10	76	40,2	12	212	212	0
30	21	262	112,56	12	653	683	0,0459
31	2	26	13,11	14	88	92	0,0455
32	1	55	18,88	14	163	163	0
33	0	45	17,08	14	162	162	0
34	0	49	25,23	14	209	209	0
35	1	42	16,89	15	166	171	0,0301
36	0	41	20,52	16	207	216	0,0435
37	13	153	74,53	16	513	513	0
38	2	99	44,42	16	513	513	0
39	4	117	61,22	18	471	507	0,0764
40	0	46	13,81	18	130	143	0,1

Последние две особенности ограничивают допустимую размерность задачи, которая решается с помощью ЭВМ за приемлемое время. В частности, при тестировании программы, реализующей алгоритм динамического программирования, на ЭВМ с процессором частотой 1,5 ГГЦ и оперативной памятью 256 МБ, удавалось решать задачу коммивояжера только для транспортных сетей, содержащих не более 45 пунктов.

3.3. Модифицированный метод динамического программирования и оценка его эффективности

В данной работе предлагается модифицированный метод динамического программирования, позволяющий, с одной стороны, уменьшить объем сохраняемых данных и количество вычислений, а с другой – повысить точность результата.

При разработке модифицированного метода динамического программирования было рассмотрено несколько вариантов модификации.

Для описания сути первого варианта рассмотрим ситуацию, когда на k-й итерации не удается построить маршрут $M_{k,i}$ на основании маршрутов предыдущей итерации.

Пример такой ситуации показан в предыдущем подразделе для транспортной сети, приведенной на рис. 3.1. На итерации $k = 5$ невозможно построить маршрут для пункта 1 на основании маршрутов предыдущей итерации, так как все маршруты итерации $(k-1) = 4$ содержат данный пункт.

В исходном методе динамического программирования допускается построение маршрута на основании итераций $k-2,\ k-3,\ ...,\ 1$, если в этом возникает необходимость.

Для уменьшения затрат ресурсов ЭВМ можно ввести ограничения на количество номеров итераций, к которым осуществляется возврат. При данном ограничении с большей вероятностью появляются пункты, для которых на отдельных итерациях станет невозможным построить маршрут.

В качестве примера снова рассмотрим транспортную сеть, показанную на рис. 3.1. Как уже было сказано, при решении задачи

коммивояжера методом динамического программирования на итерации с номером $k=4$ были получены маршруты $M_{4,1}=(1,7,6,2,5,0)$, $M_{4,2}=(2,3,1,6,7,0)$, $M_{4,3}=(3,2,1,6,7,0)$, $M_{4,4}=(4,1,3,2,5,0)$, $M_{4,5}=(5,3,1,6,7,0)$, $M_{4,6}=(6,1,3,2,5,0)$, $M_{4,7}=(7,1,3,2,5,0)$.

Предположим, что при построении маршрутов введено следующее ограничение: возможен возврат не более чем на одну итерацию. То есть при построении маршрутов на k-й итерации можно использовать только маршруты, полученные на итерациях $k-1$ и $k-2$.

Как уже было показано, для транспортной сети на рис. 3.1 на 5-й итерации при данном ограничении можно построить маршруты для всех пунктов, включая пункт с номером 1. В конце выполнения 5-й итерации были получены маршруты: $M_{5,1}=(1,3,5,2,6,7,0)$, $M_{5,2}=(2,5,3,1,6,7,0)$, $M_{5,3}=(3,1,7,6,2,5,0)$, $M_{5,4}=(4,2,3,1,6,7,0)$, $M_{5,5}=(5,2,3,1,6,7,0)$, $M_{5,6}=(6,7,1,3,2,5,0)$, $M_{5,7}=(7,6,1,3,2,5,0)$.

Очевидно, что все маршруты, построенные на итерации $k=5$ также содержат пункт 1.

При выполнении итерации $k=6$ с учетом заданного ограничения возможно использовать только маршруты, полученные на итерациях $(k-2)=4$ и $(k-1)=5$.

Таким образом, оказывается невозможным построить маршрут, который начинается в пункте 1, проходит через 6 пунктов и оканчивается в пункте 0.

Оценим точную верхнюю границу количества просматриваемых маршрутов $Q_м$. На последней, n-й, итерации ко всем построенным маршрутам присоединяется пункт 0. Точная верхняя граница количества маршрутов, просматриваемых на данной итерации, равна n ($Q_{м(n)}=n$). Количество маршрутов, просматриваемое на всех остальных итерациях, зависит от ограничений на повторный анализ

маршрутов предшествующих итераций. Таким образом, точная верхняя граница общего количества просматриваемых маршрутов

$$Q_м = Q_{м(1)} + Q_{м(2)} + ... + Q_{м(n-1)} + Q_{м(n)} = \sum_{k=1}^{n-1} Q_{м(k)} + n . \quad (3.5)$$

Оценим числа, ограничивающие количество просматриваемых маршрутов $Q_{м(1)}$, $Q_{м(2)}, ...$, $Q_{м(n-1)}$ на итерациях $1, 2, ..., n-1$, при различных ограничениях на использование предшествующих итераций.

При возможности использования на k-й итерации только маршрутов $(k-1)$-й итерации при выборе маршрута для каждого из n пунктов просматривается не более чем $n-1$ маршрут предыдущей итерации. Это число одинаково для любой итерации, то есть

$$Q_{м(k)} = n-1, \quad k = 1, 2, ..., n-1 . \quad (3.6)$$

Таким образом, за всю реализацию алгоритма вычисляются и сравниваются длины не более чем

$$Q_м = n(n-1)^2 + n \quad (3.7)$$

маршрутов с различным количеством пунктов.

Предположим, что на k-й итерации возможно использование маршрутов $(k-2)$-й итерации. В этом случае построение начинается с анализа маршрутов итерации $k-1$. Для каждого из n пунктов анализируется $n-1$ маршрут. После этого может возникнуть необходимость построения альтернативных маршрутов на основании $(k-2)$-й итерации. Пусть такая необходимость возникла для пункта i. Альтернативные маршруты мы будем строить для пунктов $1, 2, ..., i-1, i+1, ..., n$. Общее количество этих пунктов равно $n-1$. Для каждого такого пункта j (из числа указанных) целесообразно просмотреть $n-2$ маршрута, так как остальные два из обще-

го количества обязательно начинаются с пунктов i или j. Получаем количество просмотренных маршрутов $(k-2)$-й итерации $(n-1)(n-2)$. Всего таких маршрутов может оказаться $(n-1)$. Следовательно, для пункта i может быть просмотрено $(n-1)^2(n-2)$ маршрутов.

Число пунктов, для которых требуется построение альтернативных маршрутов, может быть различным для разных итераций. Велика вероятность, что на начальных итерациях не будет ни одного такого пункта, а на заключительных их будет несколько. Для приведенной на рис. 3.1 транспортной сети на итерациях 1 – 4 таких пунктов не было. На итерации 5 альтернативные маршруты потребовалось строить для одного пункта, на шестой – для трех. Число $(n-1)^2(n-2)$ является явно завышенным, так как далеко не для каждого из $(n-1)$ пунктов удается построить альтернативный маршрут.

Будем считать, что в среднем на каждой итерации необходимо построение альтернативных маршрутов для одного пункта. Тогда верхняя граница количества просматриваемых на k-й итерации маршрутов

$$Q_{м(k)} \approx (n-1)n + (n-1)^2(n-2). \tag{3.8}$$

С учетом того, что общее число итераций равно $(n-1)$, ориентировочно осуществляется просмотр не более

$$Q_м = (n-1)^3(n-2) + n(n-1)^2 + n \tag{3.9}$$

маршрутов.

При сокращении количества итераций, к которым допускается возврат, снижается также и точность полученных результатов. Поэтому для повышения точности предлагается наряду с уменьшением количества сохраняемых данных увеличить количество просмотров вариантов на небольшом количестве итераций.

Опишем k-ю итерацию предлагаемой модификации для повышения точности полученного решения. Пусть необходимо выбрать маршрут минимальной длины, начинающийся в пункте i и заканчивающийся в пункте 0. Как и в методе динамического программирования, к моменту выполнения данной итерации были построены и сохранены маршруты $M_{1,1}, \ldots, M_{1,n}$; $M_{2,1}, \ldots, M_{2,n}; \ldots;$ $M_{k-1,1}, \ldots, M_{k-1,n}$. Некоторые из этих маршрутов включают пункт i, другие его не включают. Допустим, маршруты $M_{k-1,j_1}, \ldots, M_{k-1,j_l}$ включают в себя пункт i, причем $j_1, \ldots, j_l \neq i$. Построим альтернативные маршруты $M'_{k-1,j_1}, \ldots, M'_{k-1,j_l}$, для которых выполняется условие

$$f\left(M'_{k-1,j_g}\right) = \min_{\substack{1 \le h \le n \\ h \ne j_g \\ i \notin M_{k-2,h}}} \left\{ d_{j_h h} + f\left(M_{k-2,h}\right) \right\}. \qquad (3.10)$$

Далее, рассмотрим маршруты, полученные на $(k-2)$-й итерации. Заменим те из них, в которых встречается пункт i, маршрутами минимальной длины, в которых пункт i отсутствует. При этом маршрут с минимальной длиной определяем с помощью выражения, аналогичного (3.10). Используя полученные маршруты, построим еще один набор альтернативных маршрутов $M''_{k-1,1}, \ldots, M''_{k-1,i-1}, M''_{k-1,i+1}, \ldots, M''_{k-1,n}$. Повторяем ту же операцию для итераций $k-3$, $k-4$ и так далее. После этого выбираем маршрут наименьшей длины, полученный при присоединении пункта i к одному из построенных маршрутов.

При возврате ко всем предшествующим итерациям, включая первую, количество вычислений при выполнении данной модификации имеет тот же порядок, что и количество вычислений при помощи метода полного перебора. Для сокращения вычислений предлагается ограничиться возвратом *не более чем на 4 итерации*. При этом количество вычислений не превысит числа порядка полинома пятой степени от n.

Разработан ряд модификаций метода динамического программирования, которые по сравнению с классическим методом динамического программирования обеспечивают решение задачи коммивояжера для транспортных сетей большей размерности.

В предлагаемых модификациях заложен параметр, позволяющий при решении конкретных задач делать выбор в пользу большей размерности транспортной сети или большей точности решения. В любом случае использование модификаций приводит к более качественным решениям, чем классический метод.

3.4. Тестирование модифицированного метода динамического программирования

Для выявления точности получаемых решений и максимально допустимой размерности в задачах коммивояжера, которые достижимы для предлагаемых модификаций метода динамического программирования, было проведено их тестирование. Тестирование производилось на ЭВМ с процессором частотой 1,5 ГГЦ и оперативной памятью 256 МБ. При тестировании были составлены маршруты коммивояжера для 40 различных задач размерностью от 8 до 18 пунктов (табл. 3.4).

В табл. 3.5 приведены результаты тестирования позволяющие сравнить модификации метода динамического программирования с методом «ветвей и границ» по точности получаемых решений.

В табл. 3.6 приведены максимальные размерности задач, которые удавалось решить на ЭВМ с указанными выше характеристиками за приемлемое время. Расстояния между пунктами выбирались случайным образом.

Во всех таблицах используется следующие обозначения для модификаций метода динамического программирования:

МДП1 – модифицированный метод динамического программирования с возвратом не более чем на 1 шаг для *недостижимых по результатам предыдущей итерации* пунктов;

МДП2 – модифицированный метод динамического программирования с возвратом не более чем на 2 шага для *недостижимых по результатам предыдущей итерации* пунктов;

Таблица 3.5 – Отклонение результатов, полученных при помощи модификаций метода динамического программирования, от результатов, полученных при помощи метода «ветвей и границ»

Модификация метода	МДП1	МДП2	МДП3	МДП4	МДП5	МДП6	МДП7	МДП8	МДП9	МДП10
Среднее отклонение, %	5,02	3,97	3,59	3,11	4,39	3,09	2,73	1,2	1,62	1,81
Максимальное отклонение, %	22,9	22,85	22,85	21,84	23,73	16,85	20,85	10	10	14,86

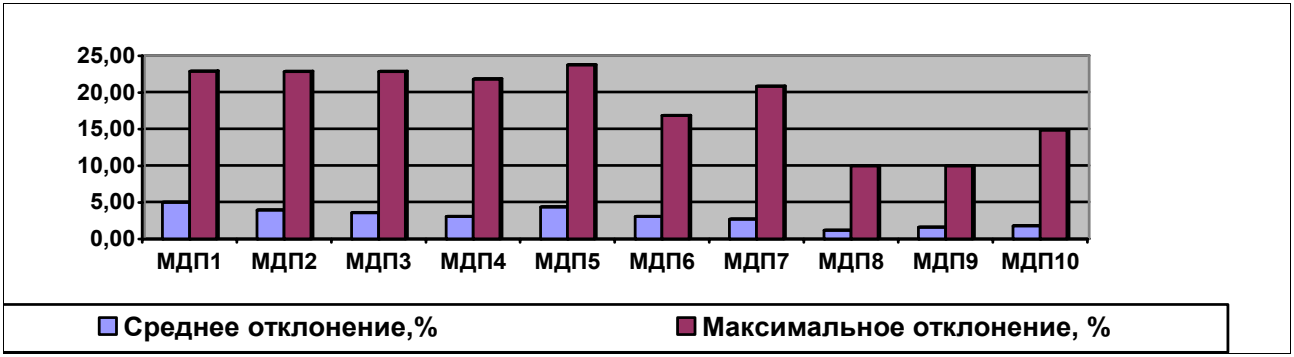


Рис. 3.3 Диаграмма относительных показателей отклонения результатов решения задачи коммивояжера, полученных при помощи модификаций метода динамического программирования, от оптимального решения

Таблица 3.5 – Максимальное количество пунктов в задачах, которые удалось решить за приемлемое время при тестировании

Модификация метода	МДП1	МДП2	МДП3	МДП4	МДП5	МДП6	МДП7	МДП8	МДП9	МДП10
Максимальное количество пунктов в задаче	120	110	95	80	100	95	90	80	75	65

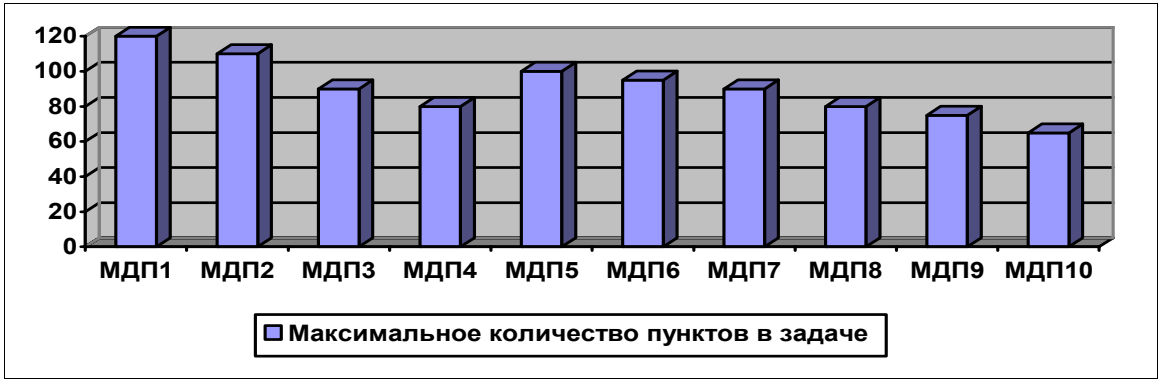


Рис. 3.4 – Диаграмма допустимой размерности задачи коммивояжера для различных модификаций метода динамического программирования

МДП3 – модифицированный метод динамического программирования с возвратом не более чем на 3 шага для *недостижимых по результатам предыдущей итерации* пунктов;

МДП4 – модифицированный метод динамического программирования с возвратом не более чем на 4 шага для *недостижимых по результатам предыдущей итерации* пунктов;

МДП5 – модифицированный метод динамического программирования с возвратом на 1 шаг для *всех* пунктов;

МДП6 – модифицированный метод динамического программирования с возвратом на 2 шага для *всех* пунктов;

МДП7 – модифицированный метод динамического программирования с возвратом на 3 шага для *всех* пунктов;

МДП8 – модифицированный метод динамического программирования с возвратом на 4 шага для *всех* пунктов;

МДП9 – модифицированный метод динамического программирования с возвратом на 5 шагов для всех пунктов;

МДП10 – модифицированный метод динамического программирования с возвратом на 6 шагов для всех пунктов.

Сравнительный анализ показывает, что наиболее выгодной с точки зрения точности является модифицированный метод динамического программирования с возвратом на 4 шага для всех пунктов (МДП8 в таблицах). Данная модификация позволяет решать задачи достаточно большой размерности. Модифицированный метод динамического программирования с возвратом на 3 шага для всех пунктов; обеспечивает ту же точность, что и метод динамического программирования, но в то же время позволяет решать задачи с удвоенным числом пунктов. При тестировании метода «ветвей и границ», который на данный момент считается самым точным и чаще всего используется, выяснилось, что на ЭВМ с выше указанными техническими параметрами можно решить задачу коммивояжера для транспортной сети, в которой не более 25 пунктов.

Модифицированный метод динамического программирования с возвратом не более чем на 1 шаг для недостижимых по результатам предыдущей итерации пунктов (модификация МДП1) позволяет решать задачи для транспортных сетей наибольшей размерности. Данным методом можно воспользоваться при необходимости полу-

чить грубую оценку длины маршрута коммивояжера для транспортной сети большой размерности.

Следует также отметить, что модифицированный метод динамического программирования с возвратом не более чем на 1 шаг для недостижимых по результатам предыдущей итерации пунктов (модификация МДП1) полезен для использования в качестве составляющих в алгоритмах, в которых встречается многократное решение задачи коммивояжера как часть решения общей задачи.

То же самое можно сказать о модифицированном методе динамического программирования с возвратом не более чем на 1 шаг для всех пунктов (модификация МДП5).

Таким образом, предлагаемую модификацию метода динамического программирования МДП5 можно считать достаточно точной и пригодной для решения задачи коммивояжера большой размерности. Описанная модификация позволяет решать задачи большой размерности за счет отказа от высокой точности, и наоборот, задавать более высокую точность решения для транспортных сетей со сравнительно небольшим количеством пунктов.

* * *

Алгоритмы предлагаемых модификаций метода динамического программирования реализованы в виде единой программы Traveling_Salesman_DP.exe. В качестве исходных данных выступают расстояния между пунктами транспортной сети. Как и в случае с задачей разбиения транспортной сети на районы, не обязательно, чтобы изначально была задана длина пути между любыми двумя пунктами. Для реальных транспортных сетей, в которых расстояния заданы не для всех пар пунктов, в программе есть специальная опция «Оптимизировать», с помощью которой при необходимости вычисляются длины кратчайших маршрутов между пунктами. Построение маршрутов реализовано с использованием матричного метода.

В качестве результата программа выдает наименьшую длину маршрута, проходящего через все пункты сети ровно по одному разу, и последовательность пунктов в маршруте. Программа также позволяет проследить последовательностью построения маршрутов, используя пошаговое выполнение алгоритма. Более подробно программа Traveling_Salesman_DP.exe будет описана в разделе 4.

РАЗДЕЛ 4

ПРОГРАММНЫЙ ИНСТРУМЕНТАРИЙ ДЛЯ ОПРЕДЕЛЕНИЯ МАРШРУТА КОММИВОЯЖЕРА И РАЗБИЕНИЯ ТРАНСПОРТНОЙ СЕТИ НА РАЙОНЫ

В данном разделе описываются программы *SlsManDProg.exe* и *Partition.exe*, реализующие модификации соответственно метода динамического программирования для решения задачи коммивояжера большой размерности и метода «ветвей и границ» для разбиения транспортной сети на районы. Названные программы являются практической реализацией алгоритмов описанных в третьем и втором разделах монографии.

Результаты тестирования программы *SlsManDProg.exe* позволяют судить о средней точности решения, которое можно получить при помощи разработанной модификации метода динамического программирования. На основании проведенного тестирования можно сделать вывод, что разработанная модификация метода динамического программирования применима при решении задачи коммивояжера для транспортных сетей больших размерностей. Высокое быстродействие программы *SlsManDProg.exe* позволяет применять ее в качестве подпрограммы в реализациях алгоритмов, в которых решение задачи коммивояжера является промежуточным действием.

Результаты тестирования программы *Partition.exe* дают возможность сделать вывод о свойствах разбиения транспортной сети на районы, полученного при помощи разработанной модификации метода «ветвей и границ».

После выполнения разбиения при помощи программы *Partition.exe* сразу становится известной длина маршрута коммивояжера, которая позволяет судить о денежных и временных затратах на транспортные перевозки в данном районе.

Использование для оценки «размеров» районов длины маршрута коммивояжера позволяет судить о площади, которую занимает район, а соотношение длин маршрутов коммивояжера – о соотношении площадей, занимаемых районами.

4.1. Программная реализация модифицированного метода динамического программирования для решения задачи коммивояжера

Программа предназначена для определения маршрута коммивояжера минимальной длины в произвольно заданной транспортной сети. Данную программу целесообразно использовать при построении маршрута коммивояжера для транспортной сети с количеством пунктов свыше 25 (для более мощных современных ПК – 40 или 50 пунктов). Верхняя граница применимости программы также зависит от мощности используемого ПК и составляет от 170 до 200 пунктов.

Как известно, маршрут коммивояжера проходит через все пункты транспортной сети ровно один раз и является кольцевым, то есть учитывает также возврат из последнего пункта в первый, с которого маршрут начинался.

Название программы – *SlsManDProg.Exe*.

Программа написана в среде программирования *Delphi* 5 и предназначена для работы под операционной системой Windows-95 (-98, -XP, -*NT* -2000 и т.д.). Никакого дополнительного программного обеспечения для запуска и выполнения программы не требуется. Редактирование и рекомпиляция программы выполняется в среде программирования *Delphi* 5 и более поздних ее версиях.

При помощи программы *SlsManDProg.exe* строится маршрут коммивояжера минимальной длины и определяется его длина. Программа допускает варьирование параметров, позволяющих увеличить допустимую размерность транспортной сети за счет отказа от точности. Наименьшее среднее отклонение искомого маршрута, получаемого в результате работы программы, от оптимального маршрута составляет 1,1%. С такой точностью получают решение для транспортной сети, содержащей не более 80 пунктов.

Программа позволяет строить маршрут коммивояжера для транспортной сети, содержащей до 170 пунктов. Однако в этом слу-

чае отклонение получаемого решения от оптимального в среднем составляет 7%.

Построение оптимального маршрута осуществляется данной программой в соответствии с алгоритмом модификации метода динамического программирования, описанного в третьем разделе данной работы.

Задача коммивояжера решается только при условии, если в транспортной сети имеется возможность построить маршрут между любыми двумя пунктами. Граф, построенный на основании такой транспортной сети, должен быть связным.

В качестве исходных данных вводятся длины путей между парами пунктов транспортной сети, для которых существует непосредственная транспортная связь. В случае, если транспортная сеть такова, что непосредственная транспортная связь существует не для всех пар ее пунктов, программа позволяет построить маршрут минимальной длины для каждой пары пунктов, не связанных между собой непосредственно.

Расстояния между пунктами транспортной сети должны быть заданы целыми числами. Среднее расстояние между парами пунктов не должно превышать числа 2147473647/(n + 1), где n – количество пунктов транспортной сети.

Алгоритм программы:

1) Запуск программы

При запуске программы на экране дисплея появляется основное окно программы, представленное на рис. 4.1. На нем присутствует главное меню, а также закладки *Матрица расстояний*, *Маршруты*, *Маршрут возврата*, *Сохраненные шаги, Оптимальный маршрут*. Закладки представляют собой страницы (*Pages*) элемента *PageControl*. На каждой из закладок находятся одна или две таблицы.

В таблице на закладке *Матрица расстояний* хранятся данные о длине пути между пунктами транспортной сети.

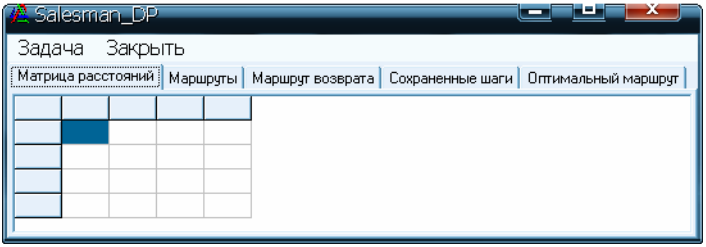


Рис. 4.1 – Основное окно программы *SlsManDProg.exe*

В таблицы на закладках *Маршруты* и *Маршрут возврата* заносятся промежуточные маршруты, получаемые в результате реализации модифицированного метода динамического программирования.

В таблице на закладке *Сохраненные шаги* запоминаются маршруты, генерируемые на каждой из итераций.

На закладке *Оптимальный маршрут* помещаются конечные результаты – длина маршрута коммивояжера наименьшей длины и последовательность пунктов в маршруте.

<u>2) Выполнение процедур, запускаемых из главного меню основного окна программы</u>

Выполнение процедур для решения задачи коммивояжера рекомендуется осуществлять в следующей последовательности:

2.1) Сначала для загрузки параметров новой транспортной сети следует выбрать в главном меню пункт *Задача/Новая* (рис. 4.2).

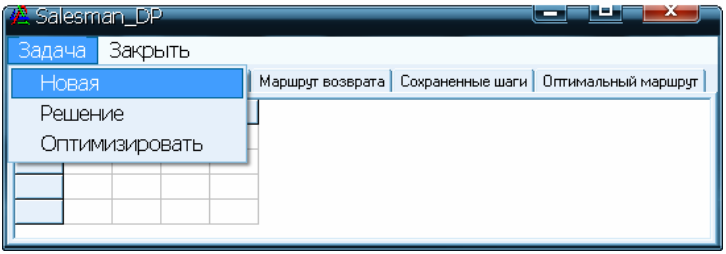


Рис. 4.2 – Выбор пункта меню для загрузки новой задачи

После выбора пункта *Задача/Новая* на экране дисплея появляется диалоговое окно для задания исходных данных задачи (рис. 4.3).

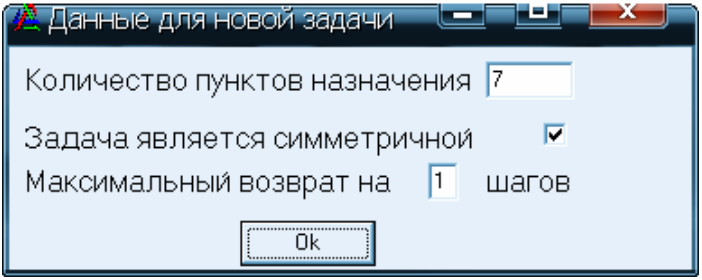


Рис. 4.3 – Диалоговое окно ввода параметров транспортной сети и управления точностью решения

В поле ввода «Количество пунктов назначения» задается количество пунктов транспортной сети. В случае, если расстояния между любыми двумя пунктами в прямом и обратном направлениях движения совпадают, необходимо поставить галочку возле поля *Задача является симметричной*. Поле *Максимальный возврат* позволяет регулировать точность получаемого решения.

В табл. 4.1 приводится зависимость точности получаемого решения от количества шагов возврата. В ней также указано максимальное количество пунктов транспортной сети, для которой можно найти решения с данной точностью.

Максимальное количество пунктов зависит не только от количества шагов возврата, но и от параметров ЭВМ, на которой работает программа. Здесь и далее приводятся результаты тестирования на ЭВМ с частотой процессора 1,50 ГГц и ОЗУ 256 Мб.

Задавать количество шагов возврата больше 4 не рекомендуется, так как при этом не только уменьшается допустимое количество пунктов в транспортной сети, но и теряется эффект повышения точности, то есть уменьшения среднего отклонения от оптимального результата.

Таблица 4.1 – Зависимости среднего относительного отклонения и максимального количества пунктов в транспортной сети от количества шагов возврата

Максимальное количество шагов возврата	Среднее отклонение от оптимального результата	Максимальное количество пунктов
0	7,00%	170
1	4,39%	100
2	3,09%	95
3	2,73%	90
4	1,20%	80
5	1,62%	75
6	1,81%	65

После ввода параметров задачи и нажатия кнопки *Ok* диалоговое окно закрывается. Производится очистка таблицы расстояний, которая находится на закладке *Матрица расстояний*, от данных предыдущей задачи и задание размеров таблиц, соответствующих параметрам решаемой транспортной сети.

Далее следует заполнить таблицу на закладке «Матрица расстояний». Будем считать, что транспортная сеть содержит n пунктов и все пункты пронумерованы числами от 0 до $(n-1)$. Если все расстояния между пунктами симметричны, то есть длина пути между пунктами i и j не зависит от направления движения, для каждой пары (i, j) достаточно ввести нужное число в ячейку, стоящую в i-й строке и j-м столбце таблицы ($i, j \in \{0, ..., n-1\}, \quad i \neq j$). В этом случае при выполнении последующих действий запускается процедура *SymmetryPerform*, при помощи которой значения из заполненных ячеек копируются в ячейки, симметричные им относительно главной диагонали таблицы. В противном случае длина пути из пункта i в пункт j вводится в ячейку на пересечении i-й строки и j-го столбца, а длина пути из пункта j в пункт i вводится в

ячейку на пересечении j-й строки и i-го столбца. Теперь такой ввод необходим как для симметричных, так и асимметричных расстояний. В случае, если между некоторыми пунктами k и l транспортная связь отсутствует, вводить что-либо в соответствующие ячейки не является обязательным, $k,l \in \{0,...,n-1\}$. Все ячейки, оставшиеся пустыми, заполняются автоматически при выполнении последующих процедур. Для такого заполнения в программе предусмотрена процедура *CheckGrid*.

2.2) Пункт меню *Задача/Оптимизировать* (рис. 4.4) выбирается в случае, если заданы расстояния не для всех пар пунктов.

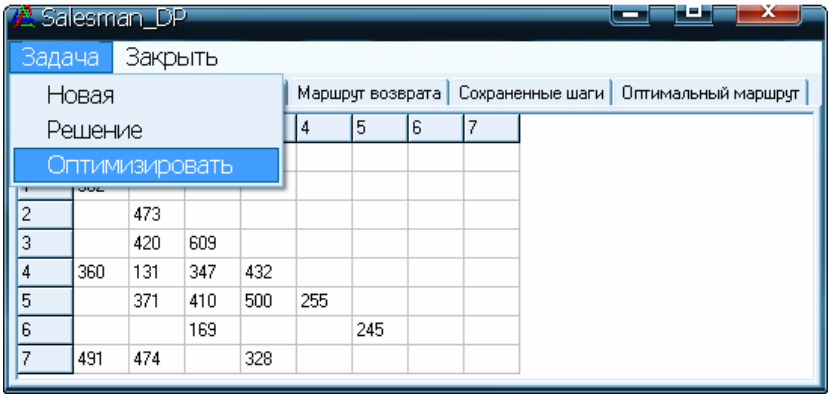


Рис. 4.4 – Выбор пункта меню для определения маршрутов минимальной длины между всеми парами пунктов

При выборе пункта *Задача/Оптимизировать* вычисляются длины маршрутов между теми пунктами, для которых непосредственной транспортной связи не существует. Построение соответствующих маршрутов выполняется при помощи встроенной процедуры *OptimNet.*

Построение маршрутов минимальной длины между пунктами, которые непосредственно между собой не связаны, осуществляется при помощи матричного метода [5]. Подробное описание указанного метода приводится в разделе 2 данной работы.

2.3) Пункт меню *Задача/Решение* (рис. 4.5) выбирается для не получения искомого маршрута коммивояжера минимальной длины.

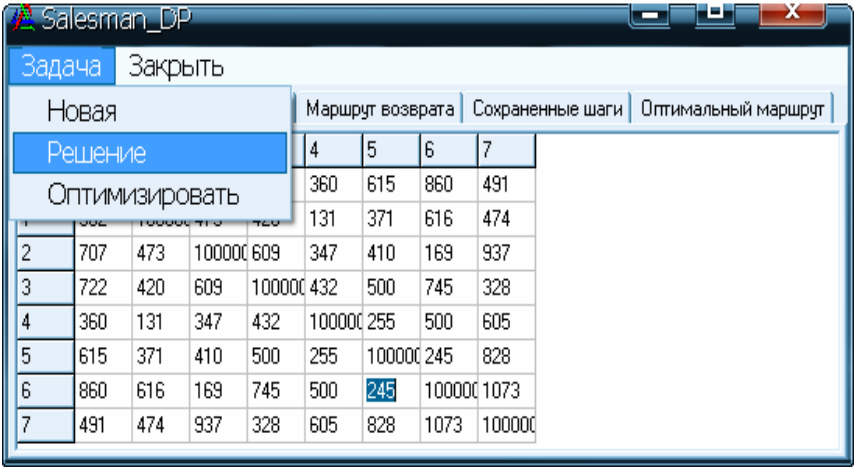


Рис. 4.5 – Выбор пункта меню для построения маршрута коммивояжера

Пункту *Задача/Решение* соответствует процедура, осуществляющая подготовку данных и таблиц к решению задачи, управление процессом решения и вывод конечного результата. Данная процедура вызывает вспомогательную процедуру *Renew*. При помощи процедуры *Renew* выполняется очистка таблиц, расположенных на закладках *Маршруты*, *Маршрут возврата* и *Сохраненные шаги*. Затем выполняется первая итерация – в верхнюю таблицу на закладке *Маршруты* вносятся маршруты. Каждый маршрут содержит два пункта, первый из которых обязательно имеет номер 0.

В нижнюю таблицу на этой же закладке вносятся длины соответствующих маршрутов. Эти же маршруты и их длины копируются в таблицу на закладке «Сохраненные шаги».

Далее $(n-1)$ раз выполняется процедура *DynStep*. В этой процедуре реализован один шаг – одна итерация метода динамического программирования.

В процессе выполнения процедуры *DynStep* производится вызов вспомогательных процедур *DynStep*1 и *DynStepBack*. Также при ее выполнении на каждом шаге осуществляется копирование полученных маршрутов в таблицу на закладке *Сохраненные маршруты*.

При помощи процедуры *DynStep*1 на основании результатов предшествующей итерации строятся маршруты $M_1, M_2, ..., M_N$, которые заканчиваются соответственно в пунктах 1, 2, …, N.

При помощи процедуры *DynStepBack* генерируются альтернативные маршруты.

Процедура *DynStepBack* требует введения двух параметров *DepthBack* и *iNeedBack*. Данная процедура строит альтернативные маршруты для пункта с номером *iNeedBack*. На основании маршрутов, полученных на итерации с номером на *(DepthBack*+1*)* меньше текущей, строятся маршруты, которые не содержат пункт *iNeedBack*. Количество пунктов в этих маршрутах постепенно доводится до величины, равной количеству пунктов в маршрутах предыдущей итерации. На основании полученных альтернативных маршрутов строится маршрут $M'_{iNeedBack}$ наименьшей длины, оканчивающийся пунктом *iNeedBack*. Если при помощи процедуры *DynStep*1 маршрут $M_{iNeedBack}$ получить не удаётся, то есть длина маршрута $M'_{iNeedBack}$ меньше длины маршрута $M_{iNeedBack}$, то маршрут $M'_{iNeedBack}$ сохраняется как результат выполнения текущей итерации. В противном случае сохраняется маршрут $M_{iNeedBack}$.

Результатом циклического выполнения процедуры *DynStep* являются маршруты, начинающиеся с пункта 0 и проходящие через все пункты транспортной сети ровно по одному разу. После этого к каждому из них присоединяется пункт 0, и определяются длины построенных маршрутов. Среди полученных кольцевых маршрутов выбирается тот, который имеет наименьшую длину. Список пунктов этого маршрута заносится в таблицу на закладке *Оптимальный маршрут*. Также на эту закладку заносится его длина.

4.2 Описание программы, реализующей алгоритм разбиения транспортной сети на районы

Программа предназначена для получения разбиения транспортной сети на районы и оценки качества полученного разбиения. Рекомендуется применять программу для транспортных сетей с количеством пунктов свыше 30 при условии, что все потребители обслуживаются одним поставщиком. Все распределяемые по районам пункты – адреса потребителей однородного груза. Местонахождение поставщика (склада) не учитывается.

Программа учитывает:

- наличие естественных препятствий между пунктами в черте города (железнодорожных путей, рек и т.д.), то есть существенные различия между расстояниями по прямой и длинами реальных транспортных маршрутов;
- неравномерное распределение потребителей по территории города, то есть различную плотность потребителей.

Название программы – *Partition.exe*.

Программа написана в среде программирования *Delphi* и работает под операционной системой *Windows*-95 (-98, -*XP*, -*NT*, -2000 и т.д.). Никакого дополнительного программного обеспечения для запуска программы не требуется. Редактирование и рекомпиляция программы выполняется в среде программирования *Delphi* 5 и ее более поздних версиях.

Посредством данной программы решается следующая задача. Заданное множество пунктов транспортной сети необходимо разбить на несколько подмножеств. Каждое подмножество представляет собой множество пунктов, принадлежащих одному из районов разбиения. Для каждой пары пунктов, имеющих между собой непосредственную транспортную связь, предварительно определяется расстояние между ними.

Результатом выполнения программы являются подмножества пунктов, определяющие районы разбиения. Кроме того, программа определяет размер каждого района разбиения как длину маршрута коммивояжера.

Разбиение осуществляется таким образом, чтобы размеры всех районов незначительно отличались между собой.

В зависимости от количества пунктов в транспортной сети, которую необходимо разбить на районы, приходится варьировать точность оценки размеров получаемых районов. Связь между количеством пунктов и точностью, с которой можно решить задачу, описана ниже.

Программа позволяет получить разбиение транспортной сети, содержащей до 80 пунктов, на любое количество районов. При этом, чем больше количество районов, тем меньше времени требуется на выполнение программы.

При разбиении сети на сравнительно небольшое количество районов (2 ÷ 5) за конечное время, следует отказываться от высокой точности оценки размеров полученных районов. Допускаемое количество пунктов в транспортной сети в зависимости от заданной точности оценки приведено в табл. 4.2.

Таблица 4.2 − Связь между ограничениями на количество районов в разбиении, количеством пунктов транспортной сети и точностью оценки размеров получаемых районов

Количество районов разбиения	Максимально допустимое количество пунктов в транспортной сети при среднем отклонении оценки размеров полученных районов от оптимальной (d)				
	d=1,1%	d=2,7%	d=3,3%	d=3,6%	d=7%
2	40	100	130	150	200
3	60	130	160	180	240
4	140	180	210	240	300
7	200	230	260	280	320

В качестве исходных данных для описываемой программы задаются расстояния между всеми парами пунктов. Если между некоторыми пунктами транспортная связь не существует, следует по-

строить маршруты минимальной длины для таких пар пунктов. В программе предусмотрено выполнение данной операции.

Расстояния между пунктами транспортной сети должны быть заданы целыми числами в одинаковых единицах для всех пар пунктов. Единицы измерения расстояния следует выбирать таким образом, чтобы среднее расстояние между парами пунктов не превышало числа 2147473647/(n + 1), где n – количество пунктов транспортной сети.

Алгоритм программы:

1) Запуск основного окна программы

Основное окно программы представлено на рис. 4.6.

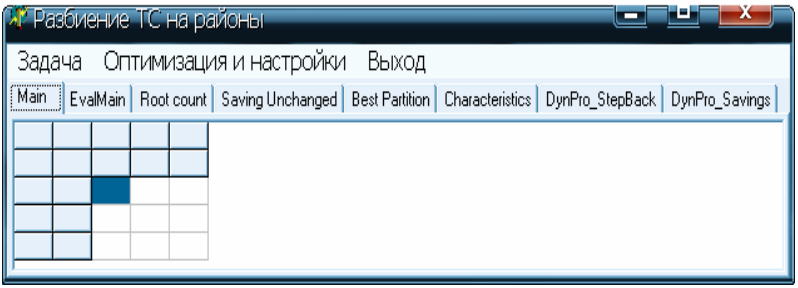


Рис. 4.6 − Основное окно программы *Partition.exe*

На основном окне программы присутствует главное меню, а также несколько закладок. На каждой из закладок расположена таблица.

Таблица, которая находится на закладке *Main*, используется для ввода исходных данных транспортной сети и хранения приведенной матрицы. В процессе формирования районов в ней также хранится промежуточная информация о составе каждого из районов.

В таблице на закладке *EvalMain* хранится матрица расстояний между пунктами. В ней, как и в таблице на закладке Main, отражает-

ся текущий состав районов на каждом этапе процесса разбиения. Таблица используется при оценке размеров полученных районов.

В таблице на закладке *Saving Unchanged* сохраняются исходные введенные данные. В таблицах на закладках *Main* и *EvalMain* исходный порядок пунктов изменяется, а на закладке *Saving Unchanged* остается неизменным.

Таблицы на закладках *RootCount, DynPro_StepBack* и *DynPro_Savings* используются при оценке размеров районов. В таблицах хранятся данные, используемые при использовании метода динамического программирования для построения маршрутов коммивояжера для каждого из районов.

В таблицах на закладках *Best Partition* и *Characteristics* после завершения процесса разбиения транспортной сети на районы фиксируются результаты, а именно: списки пунктов, относящихся к каждому из районов, длины маршрутов коммивояжера для каждого из районов и их суммарная длина.

2) Выполнение процедур, запускаемых из главного меню основного окна программы

Для выполнения задачи разбиения рекомендуется следующая последовательность действий:

2.1) Сначала осуществляют ввод данных. В поля диалогового окна (рис. 4.7), которое открывается командой главного меню *Задача/Новая* (рис. 4.8), вводятся: общее количество пунктов в транспортной сети, требуемое количество районов разбиения и данные о симметричности всех расстояний между пунктами

В программе с помощью процедуры *Starting* производится настройка размеров всех таблиц в соответствии с параметрами новой транспортной сети. Кроме того, процедура *Starting* предусматривает очистку таблиц от ненужных данных.

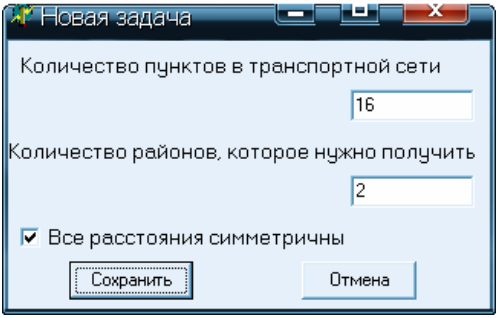


Рис. 4.7 – Диалоговое окно для ввода количества пунктов транспортной сети и требуемого количества районов

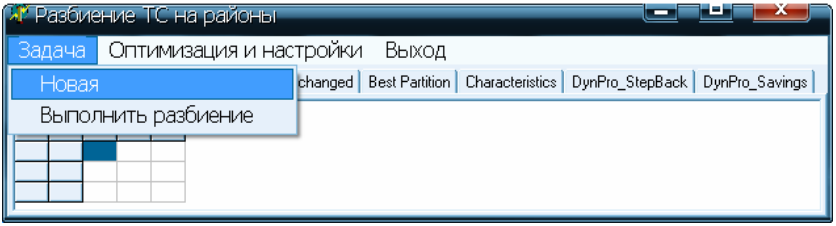


Рис. 4.8 – Выбор пункта меню для ввода данных о новой задаче

Затем в основном окне программы в таблицу на закладке *Main* вводятся расстояния между пунктами. Ввод данных осуществляется таким же образом, как и в ранее описанной программе *SlsManDProg.exe*.

2.2) После ввода исходных данных выполняется команда *Оптимизация и настройки/Оптимизировать*, которая запускает процедуру *OptimNet*. Данная команда и соответствующая ей процедура выполняются только в случае, если расстояния в предыдущем пункте алгоритма были заданы не для всех пар пунктов.

Процедура *OptimNet* строит оптимальный маршрут для тех пар пунктов, расстояние между которыми не было указано, и определяет длину построенного маршрута.

Процедура *OptimNet* идентична процедуре с тем же названием, описанной ранее для программы *SlsManDProg.exe.*

Промежуточные данные при выполнении процедуры *OptimNet* сохраняются непосредственно в таблице на закладке Main. После завершения процедуры все расстояния между пунктами и длины минимальных маршрутов заносятся также в таблицу на закладке *EvalMain*.

2.3) Непосредственное разбиение транспортной сети на районы инициируется командой главного меню *Задача/Выполнить разбиение* (рис. 4.9).

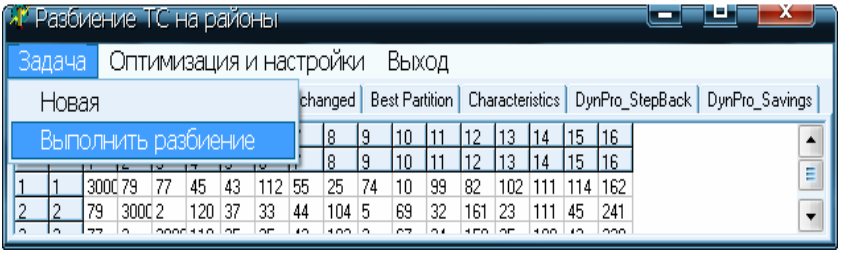


Рис. 4.9 – Выбор пункта меню для запуска процедуры разбиения транспортной сети на районы

Команда главного меню *Задача/Выполнить разбиение* осуществляет вызов процедуры *RegionForm*. Процедура выдает запрос на максимальное количество шагов возврата в алгоритме динамического программирования и на допустимое различие между районами в относительных единицах (рис. 4.10).

Как уже было сказано, максимальное количество шагов возврата в алгоритме динамического программирования регулирует точность оценки получаемого разбиения. Наиболее точным результатам соответствует максимальный возврат на 4 шага, наименее точным – на 0 шагов. Число шагов следует выбирать с учетом количества пунктов в транспортной сети. Ограничения на точность при заданном количестве пунктов приведены в табл. 4.2.

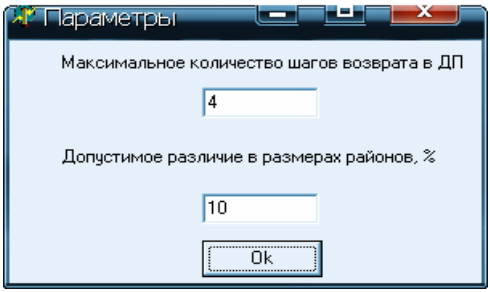


Рис. 4.10 – Форма ввода параметров для реализации разбиения

Таблица 4.3 – Влияние заданных ограничений на различие в размерах, полученных в результате тестирования районов

Тип различия	Заданное ограничение, %				
	0	5	10	25	50
Среднее различие, %	23	23,7	24,6	28,9	39,5
Максимальное различие, %	93,4	93,4	93,4	93,4	93,4
Минимальное различие, %	0,6	0,6	0,6	0,6	0,6

В силу дискретности размеров получить районы, разница в размерах которых не превышает заданную точность, иногда представляется невозможным. Поэтому задаваемая точность влияет на разницу в различии размеров районов, но не гарантирует получение указанной разницы в размерах.

В таблице 4.3 приведена зависимость между ограничением и полученными в результате тестирования различиями между размерами районов.

Согласно алгоритму, приведенному в разделе 2 данной работы, осуществляется разбиение транспортной сети на районы K способами, где K – заданное число. Среди полученных разбиений выбирается то, которое имеет наименьшую оценку. По умолчанию число

K задается равным половине общего количества пунктов в транспортной сети, т.е. $K = \frac{N}{2}$. Однако это число может быть изменено вручную при помощи пункта меню *Оптимизация и настройки/Задать количество* (рис. 4.11).

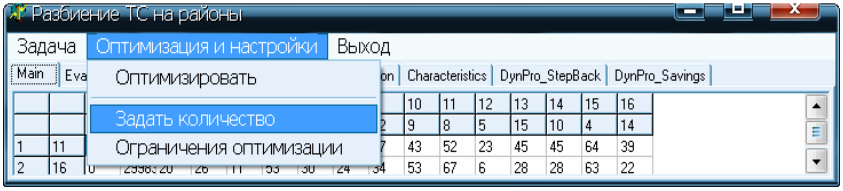


Рис. 4.11 – Выбор пункта меню для изменения количества K способов осуществления разбиения

При выборе пункта меню *Оптимизация и настройки/Задать количество* появляется диалоговое окно для задания величины K – количества вариантов разбиения (рис. 4.12). Окно позволяет задать как число вариантов, зависящее от общего числа пунктов в транспортной сети, так и любое фиксированное число вариантов.

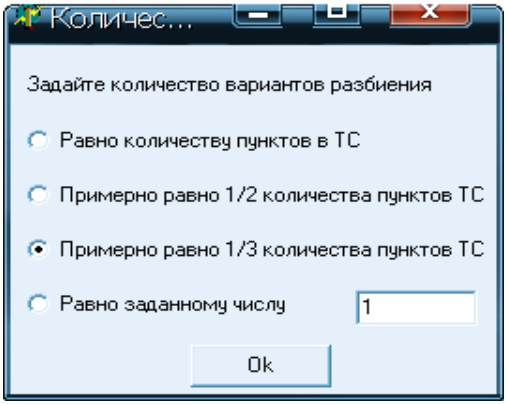


Рис. 4.12 – Задание количества вариантов разбиения

Заданное число сохраняется при разбиении всех последующих транспортных сетей. В случае, если количество вариантов разбиения было задано равным, например, $\frac{1}{3}$ от количества пунктов транспортной сети, то при разбиении новой ТС оно обновится в соответствии с ее размерностью. Если же было задано фиксированное число (выбрано *Равно заданному числу*), то оно останется неизменным. Значения по умолчанию восстанавливаются только при перезапуске программы.

Процедура *RegionForm* осуществляет циклический вызов процедуры *PerfPartition*. Перед всеми вызовами, кроме первого, осуществляет копирование данных из таблицы с закладки *Saving Unchanged* в таблицу на закладке *Main*. Оценки полученных разбиений сравниваются с наименьшей оценкой. Разбиение, которое лучше предшествующих, сохраняется. В таблице на закладке *Best Partition* запоминаются списки пунктов, которые относятся к каждому из районов наилучшего разбиения. В таблице на закладке *Characteristics* сохраняются размеры полученных районов и номер наилучшего разбиения.

В разделе 2 данной монографии был описан алгоритма модификации метода «ветвей и границ» для разбиения транспортной сети на районы. Согласно алгоритму, должно быть получено несколько разбиений, и среди них следует выбрать наилучшее. Процедура *PerfPartition* реализует часть алгоритма, соответствующую получению одного разбиения транспортной сети на районы. Процедура *PerfPartition* включает ряд вспомогательных функций и процедур. Список вспомогательных функций и процедур вместе с их назначениями приводится ниже.

Процедура *EvalMidDist* формирует вектор средних длин транспортных связей каждого из пунктов $\mathbf{L}_{cp.}$.

При помощи процедуры *Deriving* производится приведение матрицы расстояний $\mathbf{D}$. В таблице на закладке *EvalMain* сохраняются значения расстояний матрицы $\mathbf{D}$. Они используются при вычислении длин маршрутов коммивояжера. Элементы приведенной матрицы $\mathbf{D}''$ помещаются в таблицу на закладке *Main*. Числа из этой таблицы используются при формировании районов разбиения.

Функция *DetRemote* выбирает первый пункт для района. Если формируется первый район, то после выбора первого пункта района меняются местами первая строка и первый столбец, а также строка и столбец с номерами, соответствующими выбранному пункту. В дальнейшем, выбранные уже другими процедурами пункты присоединяются к району таким же образом. Вторые строка и столбец меняются местами с выбранными строкой и столбцом, затем – третьи и так далее.

Функция *NextNearestOut* определяет пункт, который является «ближайшим» к данному району. Определение пункта производится по одному из критериев: (2.11), (2.14), (2.15), (2.16). Выбор того или иного критерия происходит в зависимости от заданных параметров функции.

Функция *NearestGroup* осуществляет выбор района, к которому наиболее целесообразно присоединить текущий пункт.

Процедура *DynamicProgr* определяет длину маршрута коммивояжера для заданного района. Применив данную процедуру циклически для всех районов поочередно, можно вычислить суммарную длину маршрутов коммивояжера для всех районов. В процедуре используются вспомогательные процедуры *DynStep*, *DynStep1, DynStepBack1*. Более подробно назначение этих процедур было приведено ранее для программы *SlsManDProg.exe*.

3) Вывод полученных результатов на экран

После окончания процесса разбиения транспортной сети на районы в таблицу на закладке *BestPartition* помещаются списки пунктов, относящихся к каждому из районов. В таблице на закладке *Characteristics* находятся длины маршрутов коммивояжера для каждого из полученных районов.

4.3 Пример применения программы Partition.exe для разбиения транспортной сети на районы

Рассмотрим транспортную сеть, содержащую 61 пункт. Пункты представляют собой произвольным образом взятые точки на карте одного из районов города Харькова. Карта района города и рассматриваемые в примере пункты приведены на рис. 4.13.

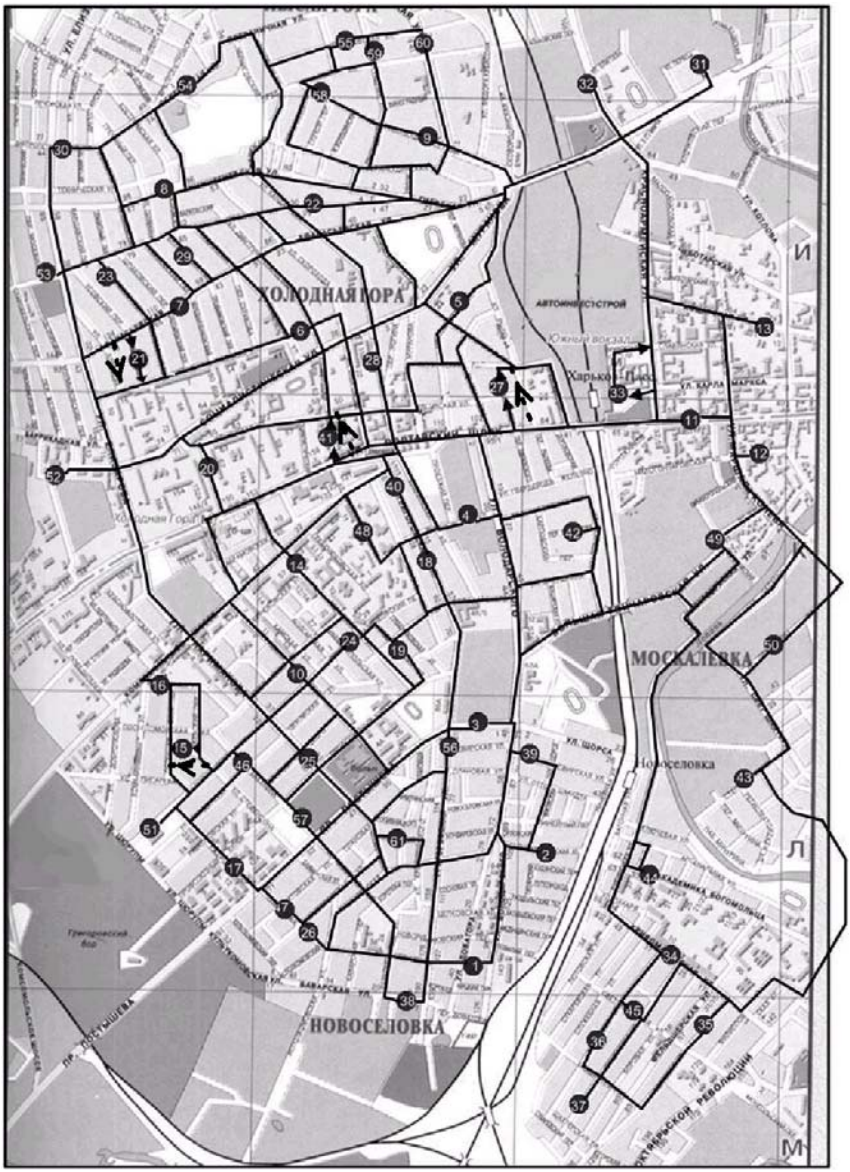


Рис. 4.13 – Карта одного из районов города Харькова

Согласно требованиям к исходным данным программы, необходимо измерить расстояния между пунктами транспортной сети. Расстояния между пунктами измеряются как длины кратчайших маршрутов по дорогам города.

Для сокращения количества измерений будем считать, что непосредственная транспортная связь существует только между теми пунктами, на маршруте между которыми не лежит никакой другой пункт.

Для общности примера будем считать, что транспортная сеть не является симметричной, то есть на некоторых участках дорог движение является односторонним. Эти участки выберем произвольным образом.

На рис. 4.13 указаны участки дорог, по которым измерялись расстояния. Те участки, на которых движение было условно задано как одностороннее, направление движения указано стрелками.

На рис. 4.14 приводится основное окно программы *Partition.exe* с уже введенными расстояниями между пунктами. Все введённые расстояния соответствуют длине участков дорог между пунктами на карте в миллиметрах. При решении данной задачи существенным является только соотношение расстояний между пунктами, поэтому масштаб карты не учитывался.

На рис. 4.15 приводится основное окно программы уже после выполнения команды *Оптимизация и настройки/Оптимизировать*. Транспортная сеть представлена в готовом к выполнению разбиения виде.

В условиях примера было выполнено разбиение транспортной сети на 2 и на 3 района. Количество рассмотренных вариантов разбиения составило примерно половину общего количества пунктов транспортной сети, то есть для обоих случаев было рассмотрено 31 вариант разбиения. Разбиение выполнялось при заданном допустимом различии районов не более 25%, средняя точность оценки полученных районов – 1,1%.

На рис. 4.16 показаны результаты, полученные программой при разбиении рассматриваемой транспортной сети на 2 района. Сформированные районы различаются своими размерами примерно на 2,8%.

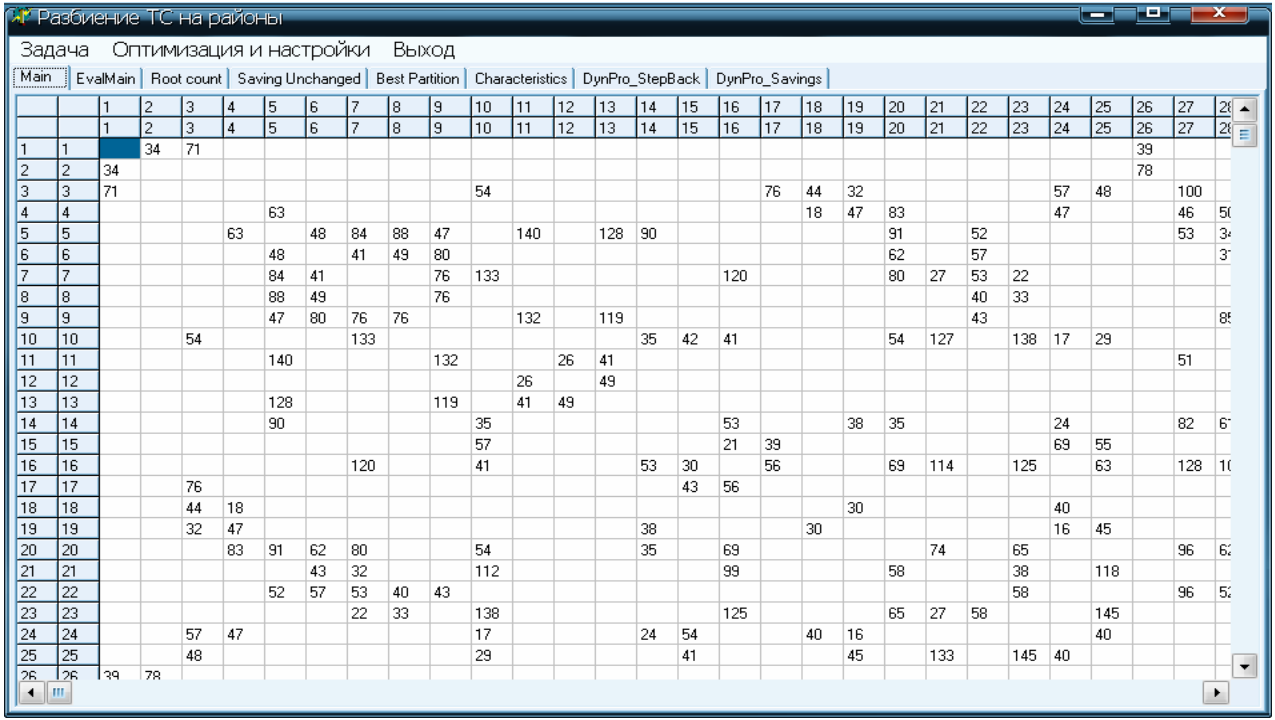


Рис. 4.14 – Основное окно программы *Partition.exe* с введенными расстояниями между пунктами

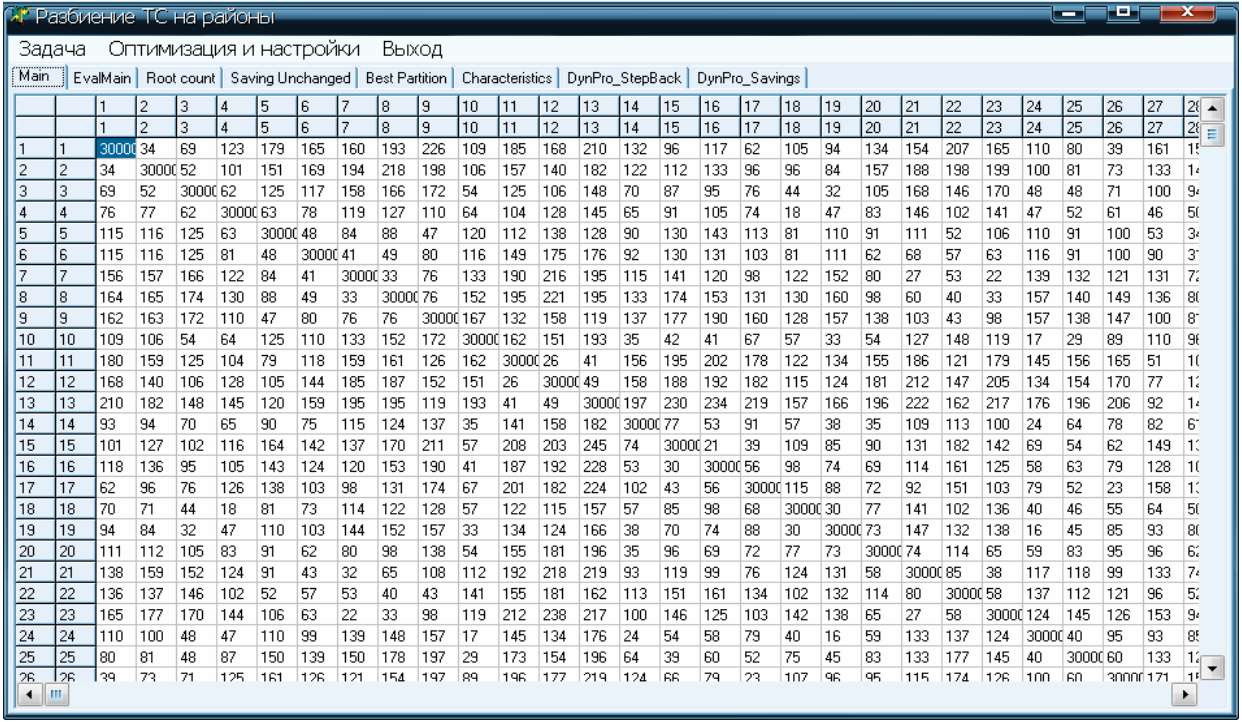
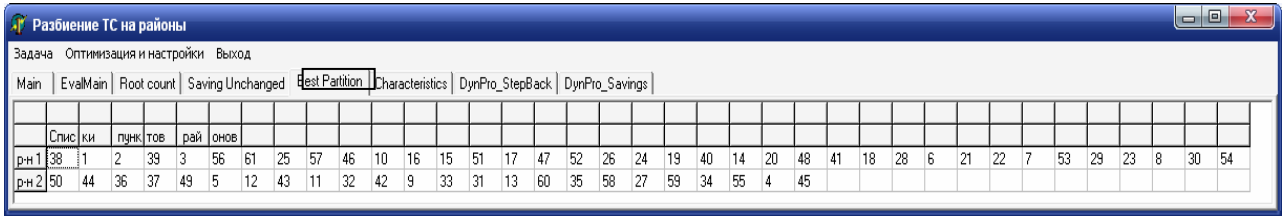
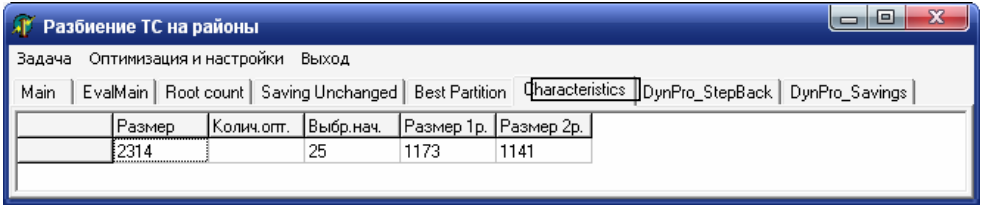


Рис. 4.15 – Основное окно программы *Partition.exe* после выполнения команды *Оптимизация и настройки/Оптимизировать*.



а) закладка *Best Partition*



б) закладка *Characteristics*

Рис. 4.16 – Результаты разбиения транспортной сети на 2 района

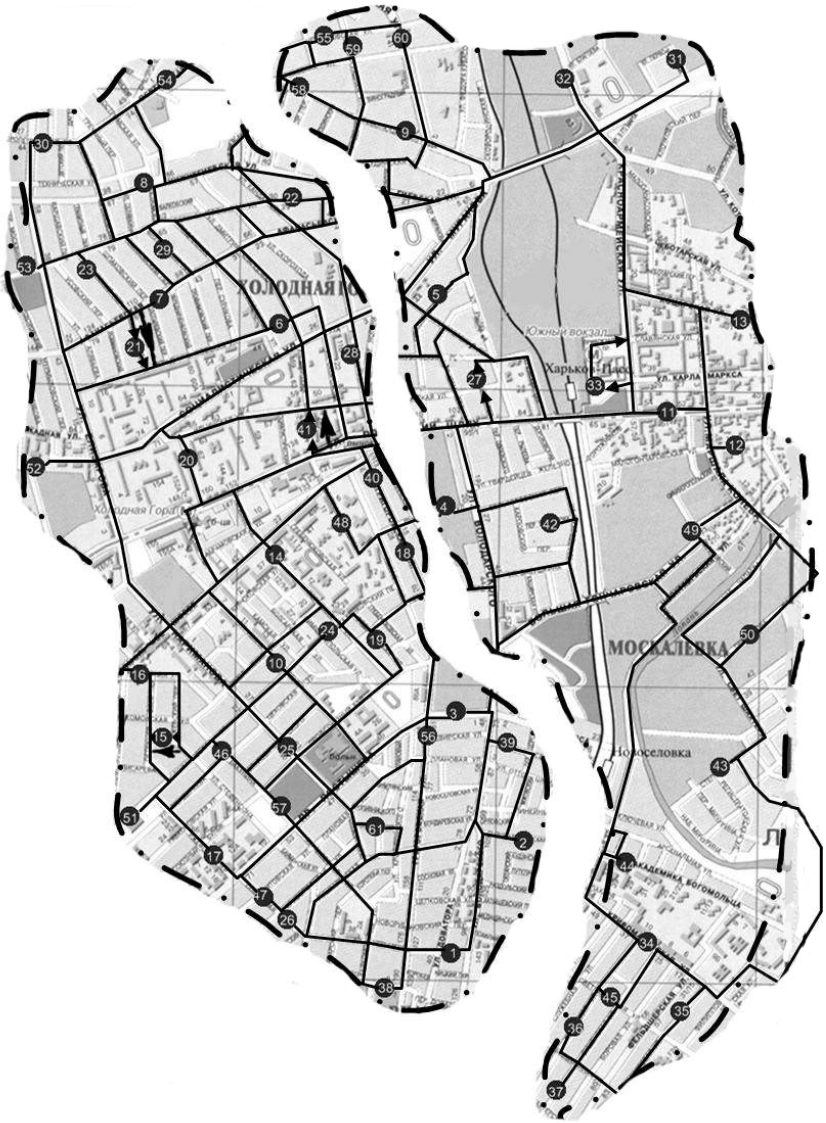


Рис. 4.17 – Результат разбиения транспортной сети на 2 района с привязкой к карте города

На рис. 4.17 результаты разбиения показаны на карте района города. Очевидно, что площади, занимаемые каждым из районов разбиения, отличаются незначительно, как и длины маршрутов коммивояжера для каждого из них.

На рис. 4.18 показаны результаты, полученные программой при разбиении транспортной сети на 3 района. Наибольшая разница между размерами районов составляет 5,1% от среднего размера района.

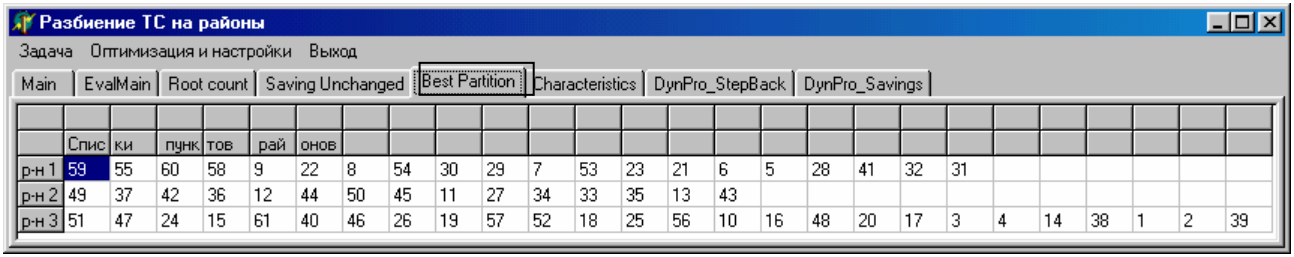
На рис. 4.19 показана карта района города Харькова после разбиения на три части в соответствии с пунктами доставки транспортной сети.

4.4. Выводы по результатам тестирования программ, реализующих модификации методов «ветвей и границ» и динамического программирования

С помощью тестирования программ, реализующих алгоритмы модификаций методов «ветвей и границ» (программа *Salesman_DP.exe*) и динамического программирования (программа *Partition.exe*), проведена оценка их практической полезности.

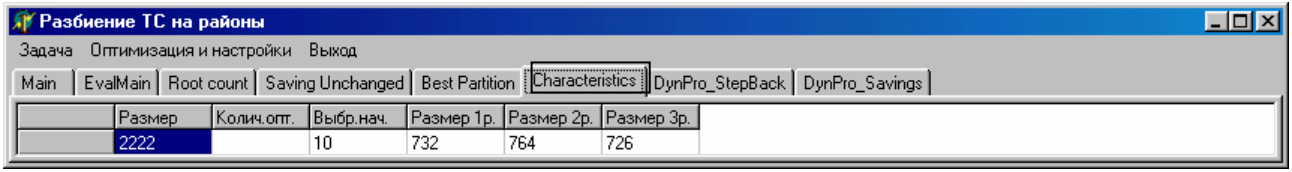
Тестирование программы *Salesman_DP.exe* показало, что при задании различных исходных данных для задачи коммивояжера средняя точность получаемых результатов составляет от 1,1% до 7%. На сравнительно устаревшем персональном компьютере решение данной задачи коммивояжера можно получить для транспортных сетей, содержащих до 170 пунктов.

В результате тестирование программы *Partition.exe* на сравнительно маломощном ПК было установлено, что разбиение на два района можно выполнить для транспортной сети, содержащей до 80 пунктов, на три района – до 90 пунктов, на четыре района – до 100 пунктов. При наличии же достаточно мощного компьютера решение задачи разбиения транспортной сети на необходимое количество районов можно получить для сетей достаточно большой размерности.



	Спис	ки	пунк	тов	рай	онов																				
р-н 1	59	55	60	58	9	22	8	54	30	29	7	53	23	21	6	5	28	41	32	31						
р-н 2	49	37	42	36	12	44	50	45	11	27	34	33	35	13	43											
р-н 3	51	47	24	15	61	40	46	26	19	57	52	18	25	56	10	16	48	20	17	3	4	14	38	1	2	39

а) закладка *Best Partition*



	Размер	Колич.опт.	Выбр.нач.	Размер 1р.	Размер 2р.	Размер 3р.
	2222		10	732	764	726

б) закладка *Characteristics*

Рис. 4.18 – Результаты разбиения транспортной сети на 3 района

Рис. 4.19 – Результат разбиения транспортной сети на 3 района с привязкой к карте города

Разработанные программы изначально были ориентированы на транспортные сети большой размерности, которые имеют место в крупных городах или регионах со значительным числом населённых пунктов. Проведенное тестирование подтвердило целесообразность их использования в инженерной практике именно для сетей большой размерности.

Тестирование также показало, что разработанный алгоритм адаптации метода «ветвей и границ» для разбиения транспортных сетей на районы позволяет производить разбиение с учетом особенностей местности и транспортных связей между пунктами.

В ходе тестирования было установлено, что разработанная модификация метода динамического программирования не только позволяет решать задачу коммивояжера большой размерности, но и решать её с высокой точностью.

* * *

Авторы будут рады, если рассмотренные модели транспортных задач и методы решения задачи коммивояжёра большой размерности или задачи разбиения транспортной сети большой размерности на заданноё число районов окажутся полезными читателям или послужат для них удачной отправной точкой в разработке новых моделей и методов.

Авторы будут благодарны всем, кто предложит новые, более универсальные средства решения рассмотренных задач или усовершенствует старые. Разработка таких средств – благодатная почва для молодых исследователей, желающих попробовать свои силы на ниве науки.

Теория транспортных перевозок ждет своих исследователей!

ИСТОЧНИКИ

1. Транспортная логистика / Под общей редакцией д-ра т.н. Л.Б. Миротина, Тамбаева Н.Э., Гудкова В.Л., М. «Экзамен», 2003. – 512 с.
2. Смехов А.А. Введение в логистику. – М.: «Транспорт», 1993. – 112 с.
3. Хоменко Л.М. Підвищення ефективності транспортних послуг в умовах трансформаційної економіки // Регіональні перспективи. – Кременчук, 2003, № 7 – 8. – С. 78 – 80
4. Лукинский В.С. Логистика // http://www.globalteka.ru/articles/docdetails/6774---.html
5. Житков В.А., Ким К.В. Методы оперативного планирования грузовых автомобильных перевозок. – М.: Транспорт, 1982. – 184 с.
6. Бочкарев А.А., Анисимова О.Н. Решение задачи оптимизации доставки мелкопартионных грузов в условиях крупного города методом локализации // Логистика сегодня. – 2008. – № 3. – С. 162 – 181.
7. Ломко Е. Автоматизация логистических процессов предприятия как один из действенных инструментов преодоления кризиса 06.11.2008 // http://consulting.1c.ru/articles-view.jsp?id=44
8. Онлайн-конференция: оптимизация транспортной логистики // http://www.ukrbiznes.com/analitic/electronic/4621.html
9. Воркут А.И. Грузовые автомобильные перевозки. Киев: «Вища школа», Головное издательство, 1986. – 447 с.
10. Шептура А.Н. Повышение эффективности автомобильных перевозок партионных грузов при переменном спросе на перевозки: Дисс. … канд. техн. наук: 05.22.01. – Харьков: ХНАДУ, 2004. – 150 с.
11. A Tutorial on Clustering Algorithms. Fuzzy C-Means Clustering // http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/cmeans.html
12. Гончаров М. Кластеризация на основе нечетких отношений // http://www.spellabs.ru/FuzzyRelationClastering.htm
13. Бочкарев А.А. Теория и методология процессного подхода к моделированию и интегрированному планированию цепи поставок: Автореф. дисс. … д-ра экон. наук: 08.00.05 / Санкт-Петербург. экон. ун-т. – СПб., 2009. – 39 с.
14. Корбут А.А., Филькенштейн Ю.Ю. Дискретное программирование // Под редакцией Юдина Д.Б. – М.: Наука, 1969 – 368 с.
15. Беллман Р. Применение динамического программирования к задаче о коммивояжере. //Кибернетической сборник. – 1964. – Вып. 9. – С. 219 – 222.
16. Куликов А. Алгоритмы и структуры данных. Лекция 7: Динамическое программирование // http://logic.pdmi.ras.ru/~kulikov/simplealgscourse/lecture7-slides.pdf С. 96

17. Зарецкий Л.С. Решение задачи коммивояжера и задач развозки методом коррекции функций состояния. // Методы оптимизации перевозочного процесса на автотранспорте. – М., 1976. С. 70 – 83.

18. Литтл Дж. и др. Алгоритм решения задачи коммивояжера. // Экономика и математические методы. – 1965. - №1. – С. 94 – 107.

19. Крушевский А.В. Задача о бродячем торговце //Материалы научных семинаров по теоретическим и прикладным вопросам кибернетики. – Киев, Освіта, 1964. – 320 с.

20. Foster B.A., Ryan D.M. An integer programming approach to the vehicle scheduling problem. // Operational Research Quarterly. – 1976. – V.4, №1. – P. 367 – 384.

21. Beltrami E., Bodin L. Networks and vehicle routing for municipal waste collection. // Networks. – 1974. – V.4, №1. – P. 65 – 94.

22. Evans S.R., Norback J.P. The impact of a decision-support system for vehicle routing in foodservice supply situation. // Journal of the Operational Society. – 1985. – V.36, №4. – P.467 – 472.

23. Чалый А., Рыбак Б. Ситуационные методы планирования и управления перевозками мелкопартионных грузов. // Автомобильный транспорт. – 1983. - №9. – с. 16 – 19.

24. Christofides N., Eilon S. Algorithm for large-scale traveling salesman problem. // Operational Research Quarterly. – 1972. – V. 23, №4. – P.511 – 518.

25. Groes G. A method for solving traveling salesman problems. // Operations Research. – 1958. – V.6, №5. – P. 791 – 812.

26. Russel L. An effective heuristic for M-tour traveling salesman problem with some side condition. // Operations Research. – 1977, V. 25, №3. – P. 517 – 524.

27. Ефремов А.В. Трахтенберг Л.Б. Совершенствование планирования перевозок и их исследование. // Методы исследования операций в задачах автомобильного транспорта. – М.: МАДИ, 1983. – С. 4 – 9.

28. Gillett B., Miller L. A heuristic algorithm for the vehicle dispatch problem. // Operations Research. – 1974. – V. 22, №3. – P. 340 – 349.

29. Miller G.E., Tucker A.W., Zemlin R.A. Integer programming formulation of traveling salesman problems. // Computer Machinery. – 1960. V. 7, №4. – P. 326 – 329.

30. Clarck G., Wright J. Scheduling of vehicles from a central depot to a number of delivery points. // Operations Research. – 1964. – V. 12, №4. – P. 568 – 581.

31. Gaskell T.I. Bases for vehicle fleet scheduling. // Operations Research Quarterly. – 1967. V. 18, №2. – P. 281 – 295.

32. Golden B., Magnanti I., Nguyen H. Implementing vehicle routing algorithm. // Networks. – 1977. – V.7, №2ю – P. 113 – 148.

33. Holmes R.A., Parker R.G. A vehicle scheduling procedure based upon savings and solution perturbation scheme. // Operations Research Quarterly. – 1976. – V. 27, №1. – P. 83 – 92.

Наукове видання

САМОЙЛЕНКО Микола Іванович,
КОБЕЦЬ Анна Олександрівна

ТРАНСПОРТНІ СИСТЕМИ ВЕЛИКОЇ ВИМІРНОСТІ

Монографія
(рос. мовою)

Відповідальний за випуск – Самойленко М.І.
Технічне редагування – Кобець А.О.

Видавництво «НТМТ»
Свідоцтво про Державну реєстрацію ДК № 1748 від 15.04.2004 р.
61072, м. Харків, пр. Леніна, 58 к. 106.
Тел. 763-03-80, 763-03-72

Підписано до друку 30.08.2010.
Формат 60х84/16. Папір 80 г/м^2.
Умов.-друк. арк. – 13,25. Обл.-вид. арк. – 14,0.
Тираж 300 примірників.

Отпечатано в типографии ООО «Цифра Принт» на цифровом комплексе XEROX DocuTech 6135.
Адрес: г. Харьков, ул. Культуры, 22-Б. Телефон : (057) 7861860.